

# Chapter 2: Gaining Initial Access

---



# Review of Chapter 1

- MITRE Pre-ATT&CK = Reconnaissance & Resource Development
- Active Recon: SYN scans, DNS service probes
- Passive Recon: DNS subdomain lookups & reverse DNS
- Defensive deceptions and delay tactics

# Key Takeaways from Chapter 1

---

- Reconnaissance = attacker intelligence gathering
  - Python enables automation of recon tasks
  - Defenders can detect or mislead attackers
-

# MITRE ATT&CK: Initial Access

- Let's talk about this process again
- So we did our recon, what's next?

Reconnaissance (10)  
Resource Development (7)  
**Initial Access (9)**  
Execution (12)  
Persistence (19)  
Privilege Escalation (13)  
Defense Evasion (40)  
Credential Access (15)  
Discovery (29)  
Lateral Movement (9)  
Collection (17)  
Command and Control (16)  
Exfiltration (9)  
Impact (13)

Drive-by Compromise  
Exploit Public-Facing Application  
External Remote Services  
Hardware Additions  
Phishing (3)  
**Replication Through Removable Media**  
Supply Chain Compromise (3)  
Trusted Relationship  
**Valid Accounts (4)**

# MITRE ATT&CK: Initial Access

- Goal: Establish foothold in target environment
- Common methods? (Phishing, credential theft, removable media, etc.)

Reconnaissance (10)  
Resource Development (7)  
**Initial Access (9)**  
Execution (12)  
Persistence (19)  
Privilege Escalation (13)  
Defense Evasion (40)  
Credential Access (15)  
Discovery (29)  
Lateral Movement (9)  
Collection (17)  
Command and Control (16)  
Exfiltration (9)  
Impact (13)

Drive-by Compromise  
Exploit Public-Facing Application  
External Remote Services  
Hardware Additions  
Phishing (3)  
**Replication Through Removable Media**  
Supply Chain Compromise (3)  
Trusted Relationship  
**Valid Accounts (4)**

# Technique 1: Valid Accounts

---

- Use of legitimate credentials
  - Sources: phishing, data dumps, brute force, keylogging
  - Targets: Telnet (23), SSH (22), RDP (3389), FTP (21), etc.
-

# Sample Telnet Authentication

```
.....!.."'.#.%..%.....!.."..P.  
..."..b.....b.... B.  
.....".....'.....#..&..&..$..&..&..$..  
.....#.....'..... .9600,9600....#.bam.zing.org:  
0.0....'..DISPLAY.bam.zing.org:0.0.....xterm-  
color.....!....."  
OpenBSD/i386 (oof) (ttyp2)  
  
login: fake  
.....Password:user  
  
.....Last login: Sat Nov 27 20:11:43 on ttyp2 from bam.zing.org
```

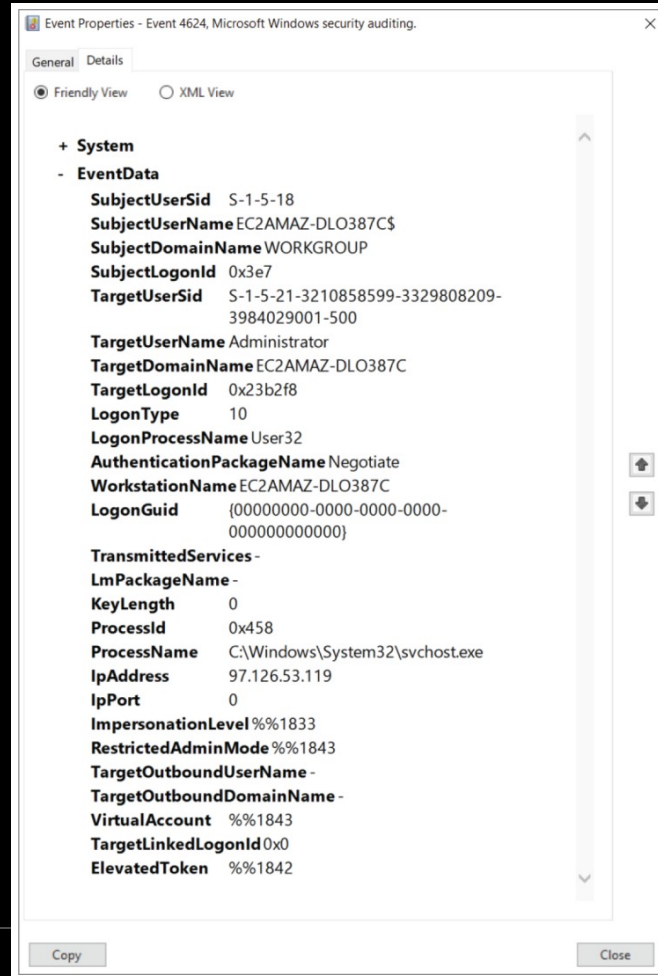
# TestDefaultCredentials.py

---

- Illustrates brute force credential testing
- Automated login attempts against SSH and Telnet
- You should always create your own list, not just rockyou.txt



# Event 2624

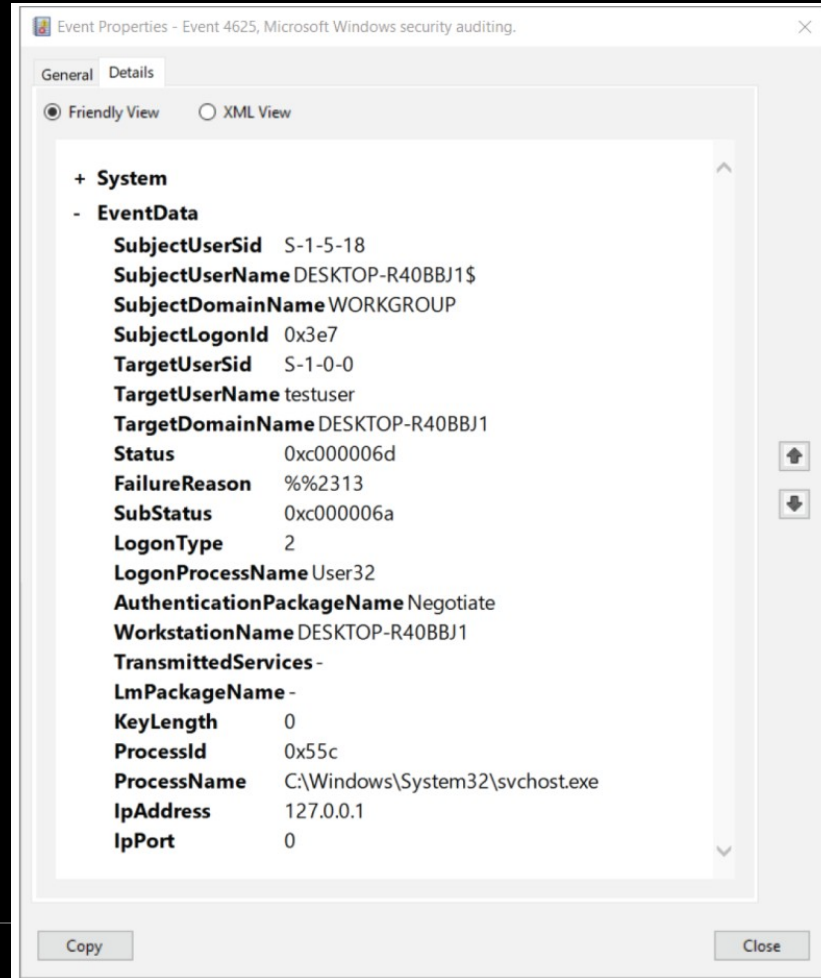


# Windows Logon Types

LOGON TYPE	LOGON TITLE	DESCRIPTION
0	System	Used only by the System account, for example at system startup.
2	Interactive	A user logged on to this computer.
3	Network	A user or computer logged on to this computer from the network.
4	Batch	Batch logon type is used by batch servers, where processes may be executing on behalf of a user without their direct intervention.
5	Service	A service was started by the Service Control Manager.
7	Unlock	This workstation was unlocked.
8	NetworkCleartext	A user logged on to this computer from the network. The user's password was passed to the authentication package in its unhashed form. The built-in authentication packages all hash credentials before sending them across the network. The credentials do not traverse the network in plaintext (also called <i>cleartext</i> ).

LOGON TYPE	LOGON TITLE	DESCRIPTION
9	NewCredentials	A caller cloned its current token and specified new credentials for outbound connections. The new logon session has the same local identity, but uses different credentials for other network connections.
10	RemoteInteractive	A user logged on to this computer remotely using Terminal Services or Remote Desktop.
11	CachedInteractive	A user logged on to this computer with network credentials that were stored locally on the computer. The domain controller was not contacted to verify the credentials.
12	CachedRemoteInteractive	Same as <code>RemoteInteractive</code> . This is used for internal auditing.
13	CachedUnlock	Workstation logon.

# Event 4625



# ValidAccountDetection.py

---

- Parse Windows Event Logs
- Event ID 4624
- Event ID 4625
- Flag multiple failed logins from same source



# Technique 2: Removable Media

---

- Malware spread via USB drives
  - Old way: Autorun
  - New way: Social engineering
  - Attackers disguise malware as documents
-

# AutorunSetup.py

- Copy malicious payload onto USB drive
- Rename it something innocent (report.exe, chrome, etc.)

Name	Date modified	Type	Size
 Autorun.inf	7/15/2021 7:44 PM	Setup Information	1 KB
 benign.exe	7/15/2021 7:44 PM	Application	7,365 KB

# AutorunDetection.py

---

- Detect USB mounted events
- Scan for executable in removable directories
- Alert on suspicious additions

# Optional Practice

---

- TestDefaultCredentials performs a credential stuffing attack against SSH and Telnet servers. Modify the code to work for other protocols as well, such as FTP or SMTP.



# Optional Practice

---

- Currently, `ValidAccountDetection` only prints out the number of failed login attempts for a particular user account. Modify the code to perform additional behavioral analytics, such as using the timestamps in log files to detect sudden bursts of login attempts.

# Optional Practice

---

- AutorunDetection determines the PID of a process using the psutil library. Modify the code to provide additional information about suspicious processes, such as their creation time or parent PID.