The Dojo of Blue: How Adversary Emulation Can Enhance Blue Team Performance

Shang-De Jiang
Cyber Security Researcher, CyCraft
Who am I

- Cyber Security Researcher @ CyCraft
- Speaker of HITCON, Black Hat USA(2020)
- UCCU Hacker Co-Founder
  - Private Cyber Security Group in Taiwan
CyCraft is an AI company that forges the future of cybersecurity resilience through autonomous systems and human-AI collaboration.
CyCraft in MITRE ATT&CK Evaluation

CyCraft Takes Significant Alerting Lead in MITRE ATT&CK® Evaluations’ Latest Round
Maturity level

Vuln Management → Penetration Testing → Blind/Internal Red Team → In Person/Continues Purple Team

Ref: Bryson Bort (scythe)
Why Adversary Emulation?

• Check detect/investigate capability
  • Can our products can detect known attack?
  • Do we need to add more detection?

• Validate SOC/Blue Team
  • Check MSSP still awake
Our Adversary Emulator Goals

- Easy to build the environment
- Continuous add new attack framework
- Enhance the investigation skills
- Make historical security event can be replay
Agenda

• Emulator Architecture
• Emulation Process Design
• Toolkit integrated
• Blue Team Performance
Architecture

- Visualization
- Management
- CLI
- DB
- Python Class API
- Metasploit Wrapper
- Empire Wrapper
- SharpSploit Wrapper
- Metasploit
- Empire
- SharpSploit

Socket + Json API
Current stage: wrappers directly talk to visualization
Infrastructure Builder

Visualization

Management

Syslog Server

1. Setup victim infra topology

2. Setup red team playbook

Infrastructure Builder

Playbook Builder

Victim Infra

VagrantAPI

1. Launch Attack

Empire Engine

Metasploit Engine

Power Sploit Engine
Code Snippet:

Management = new Management()

Endpoint m1 = new windows.Endpoint("windows 10")
Endpoint m2 = new windows.Endpoint("windows server 2016")

env_handler = Management.new_infra()
env_handler.add_machine(m1)
env_handler.add_machine(m2)
env_handler.init_infra()

operation_plan = get_available_operation_plans(env_handler)
Management.new_op_plan("empire apt3")

operation = Management.launch_operation(operation_plan, env_handler)
Other infra builder project

• Mordor Labs (https://github.com/OTRF/mordor-labs)
• attack_range (https://github.com/splunk/attack_range)
Attack Simulator

Network Environment

Attacker View

ATTACK ID: T1005 Data from Local System
ATTACK ID: T1074 Data Staged
ATTACK ID: T1105 Remote File Copy
ATTACK ID: T1158 Hidden Files and Directories
ATTACK ID: T1102 Data Compressed
Press any key to continue
Take sethc in victim B
ShellCmd: shell
wmic /node:10.99.99.102 /password:1qaz@WSX /user:eric service list
wmic /node:10.99.99.102 /password:1qaz@WSX /user:eric process list
wmic /node:10.99.99.102 /password:1qaz@WSX /user:eric startup list

ATTACK ID: T1219 Remote Access Tools
ATTACK ID: T1015 Accessibility Features
ATTACK ID: T1183 Image File Execution Options Injection
Press any key to continue
Connected
[ ] Initial server connection...
[*] Launching werkzeug_debug_rce against 192.168.122.135
[*] Waiting 30 seconds...
[*] Waiting for reverse meterpreter connection from 192.168.122.135...
[*] Waiting 30 seconds...
[*] Waiting 29 seconds...
[*] Got meterpreter connection from 192.168.122.135
[*] Waiting 28 seconds...
[*] Pwned 192.168.122.135 by exploiting werkzeug_debug_rce => session 5
[*] Finished exploiting 192.168.122.135 using werkzeug_debug_rce
[*] Waiting 27 seconds...
[*] Waiting 26 seconds...
[*] Waiting 25 seconds...
[*] Waiting 24 seconds...
[*] Waiting 23 seconds...
[*] Waiting 22 seconds...
[*] Waiting 21 seconds...
[*] Got meterpreter connection from 192.168.122.220
[*] Pwned 192.168.122.220 by exploiting october_upload_bypass_exec => session 6
[*] Finished exploiting 192.168.122.220 using october_upload_bypass_exec
[*] Connected to server

192.168.122.135
users-iMac-4.local
localhost.localdomain
Adversary Emulator

Playbook design
Playbook design
Playbook – Design Concept

• Technique – modularize the attack procedure

• Story – Enhance the blue team investigation skills

• Not just detectable technique
Playbook

APT3
https://attackevals.mitre.org/

APT29
https://attackevals.mitre.org/

Dogeza
Dogeza Playbook Scenario

<table>
<thead>
<tr>
<th>Role</th>
<th>Software and Environment</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Team</td>
<td>Kali 4.15.0, MS15-015</td>
<td>172.16.40.225</td>
</tr>
<tr>
<td>Blue Team</td>
<td>Xensor, CyCraft, CyberTotal</td>
<td>172.16.40.230</td>
</tr>
<tr>
<td></td>
<td></td>
<td>172.16.40.231</td>
</tr>
<tr>
<td>Victim A</td>
<td>Linux Ubuntu 16.04</td>
<td>172.16.40.232</td>
</tr>
<tr>
<td>Victim B</td>
<td>Windows Server 2012 R2</td>
<td>172.16.40.226</td>
</tr>
<tr>
<td>Victim C</td>
<td>Windows 10 (1607) English</td>
<td>172.16.40.227</td>
</tr>
</tbody>
</table>
Dogeza Red-Blue Team Step

• Part I – Setup & Linux Red

<table>
<thead>
<tr>
<th>Step</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Blue Team then deploys software on Victim A, B and C</td>
</tr>
<tr>
<td>2</td>
<td>Red Team use web exploit to attacks Victim A</td>
</tr>
<tr>
<td>3</td>
<td>Red Team takes privilege escalation in Victim A</td>
</tr>
<tr>
<td>4</td>
<td>Red Team implants forged ssh key for persistence</td>
</tr>
<tr>
<td>5</td>
<td>Red Team installs a kernel rootkit and hides a process in Victim A</td>
</tr>
<tr>
<td>6</td>
<td>Red Team constructs a tunnel to reach internal Victim B</td>
</tr>
</tbody>
</table>

• Part II – RT & BT Investigation

<table>
<thead>
<tr>
<th>Step</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Red Team exploits Victim B via the tunnel to implant webshell (skip, duplicated as step 3)</td>
</tr>
<tr>
<td>9</td>
<td>Red Team launch webshell of Victim B</td>
</tr>
<tr>
<td>10</td>
<td>Red Team obtains the privilege and credentials of Victim B</td>
</tr>
<tr>
<td>11</td>
<td>Red Team moves laterally to Victim C</td>
</tr>
<tr>
<td>12</td>
<td>Red Team collects sensitive documents and deploys backdoor on Victim C</td>
</tr>
<tr>
<td>13</td>
<td>Blue Team generates investigation report</td>
</tr>
</tbody>
</table>
Red Team Procedure: Step 3 Initial Access

- Use CVE-2019-9194 to exploit elFinder for www-data privilege shell
- elFinder is a famous file manager for web, and many 3rd party integration
  - Django
  - Drupal
  - Laravel
  - Widely used and directly put to public network

- CVE-2019-9194 is a command injection vulnerability in the elFinder's PHP connector.
  - High severity – remote code execution
  - Easy to launch attack – Metasploit module available
Red Team Procedure: Step 4

• Red team rises his privilege through vulnerability in chkrootkit
• CVE-2014-0476 – chkrootkit will invoke a world-writable file /tmp/update as root. Therefore if this file is modified by attacker, the root privilege can be harvested.
  • Generate and put our reverse shell in /tmp/update
  • Compare to kernel exploit, this kind of privilege escalation is more stable and easy.

```bash
upload vnsecurity/shell /tmp/update
chmod 755 /tmp/update
```
Red Team Procedure: Step 5

• Red team achieve persistence via 2 steps
  • Implant a forged ssh key into ~/.ssh/ authorized_keys
  • Modify `/etc/sudoer` to make compromised account can sudo without password

• The user is origin user in system and with the same privilege (unless not using password for `sudo`), more difficult to find out

```bash
use linux/manage/sshkey_persistence
set session 2
exploit
```
Red Team Procedure: Step 6

• Red team install rootkit to keep stealthy and prevent detection
  • Hide our meterpreter process

• In this scenario, our red team uses Retile rootkit
  • A kernel mode rootkit
  • Most famous (most starts) rootkit project in Github

```
root@ubuntu:~$ ps -a
   PID TTY      TIME CMD
 1210 pts/0  00:00:00 tmux
 2995 pts/1  00:00:00 python3
 2996 pts/2  00:00:00 ps
root@ubuntu:~$ /reptile/reptile_cmd hide 2995
Success!
root@ubuntu:~$ ps -a
   PID TTY      TIME CMD
 1210 pts/0  00:00:00 tmux
 2998 pts/2  00:00:00 ps
```
Red Team Procedure: Step 7

• Red team setup a tunnel to reach the internal web services
  • Thus the external attacker can access to internal services
  • While many IT put a lot of afford in network boundary, the security in intranet may be fragile
• In this scenario, we use socat for tunneling
  • Not really a malware
Red Team Procedure: Step 8 & 9

- Then, we move on to the Windows victims
- In reality, we need a exploit in web server to initial access to Win Server 2012
- In the demo, since web exploit is already conduct in Step 3, we would not cover the web exploit in here.
- The webshell is directly deployed in Win Server 2012
Red Team Procedure: Step 10

• Escalate privilege from IIS to system
  • Use wehshell to trigger privilege escalation
  • The privilege escalation will bring the reverse shell for menterpreter

• MS15-015/CVE-2015-0062
  • it fails to properly validate and enforce impersonation levels.
  • An attacker who successfully exploited this vulnerability could bypass impersonation-level security checks and gain elevated privileges on a targeted system.
  • This vulnerability can be exploited only in the specific scenario where the process uses SeAssignPrimaryTokenPrivilege, which is possible existed for normal processes.

• Meanwhile, Mimikatz is utilize to gain the credential of Eric in the Victim B. The retrieved credential could used to query Victim C.
Red Team Procedure: Step 11 & 12

- Red team uses several administrative tools to control Victim C.
  - Bitsadmin
  - PSEXEC
  - wmi
- Since these tools are not malicious, anti-virus rarely discovers these attacks.
- These tools are used to gain following information
  - Process list
  - Service list
  - Startup list
  - Deploy keylogger
- Red team collect top confidential information and send back to Victim B's web, then these stolen data exfiltrate via Victim A's tunnel.
Red Team Procedure: Step 12

• Red team collect top confidential information and send back to Victim B’s web, then these stolen data exfiltrate via Victim A’s tunnel.

• The collected data is compressed by a rarely used, but build-in compression tool - makecab

• The collected data is temporarily put into Recycle Bin to prevent detection
Attack Toolkit Integrated
Metasploit Integrated

• Pros
  • Great Exploit & Vulnerability resource
  • Well design session management

• Cons
  • Interactive with RPC is complicated
Empire Integrated

• Pros
  • Known PowerShell post-exploitation framework
  • Simple Agent Management design

• Cons
  • No longer being supported and development has stopped.
    • But there are other successor
Repurpose the APT malware

• Closer the real-world case

• Emulate the most APT group in the region

• APT malware usually has well-design to evade security product
APT malware - DBGPRINT

- APT Group:
  - WaterBear, Plead, BlackTech
- Since at least 2009
- Multi variant
  - Targeting security product by inject shellcode to evade detection
  - Runtime decrypt encrypted function
  - Anti-memory forensics
- Remote download dll and load
DBGPRINT stager flow

1. Check debug environment
2. Relocate function table
3. Init API from hash table
4. Get DLL
   • Remote download from C2
   • Load from local file
5. Decrypt inject & Execute in memory

Replace with integrated toolkit payload here
Blue Team Evolution
**Detect Target – PowerShell OS cred dumping**

- ATT&CK evaluation – APT29 step 6.C.1, PowerShell Dump OS credential

| 6.C.1 | Dumped password hashes from the Windows Registry by injecting a malicious DLL into Lsass.exe | powershell.exe injecting into Lsass.exe OR Lsass.exe reading Registry keys under HKLM:\SAM\SAM\Domains\Account\Users\ | Credential Dumping (T1003) |
The attack method want to detect

- PowerShell download remote script
- OS Credential Dumping via PowerShell
Detect from command line

+ System
- EventData

  SubjectUserSid  S-1-5-21-2000993884-2608570164-3450280588-1001
  SubjectUserName Nancy
  SubjectDomainName DESKTOP-K3CJE60
  SubjectLogonId  0x1ca8
  NewProcessId    0x1f44
  NewProcessName  C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
  TokenElevationType  %%%1938
  ProcessId       0x135c
Detect from process loaded library
Check PowerShell eventlog

- **EventID : 4104**

  - Creating Scriptblock text (1 of 1):
    ```powershell
    ```

- **EventID : 4103**

  - CommandInvocation(Compress-Archive): "Compress-Archive"
    - ParameterBinding(Compress-Archive): name="LiteralPath"; value="C:\Users\pbeesly\Desktop\Microsoft Edge.Ink, C:\Users\pbeesly\Favorites\Bing.url, C:\Users\pbeesly\Links\Desktop.Ink, C:\Users\pbeesly\Links\Downloads.Ink"
    - ParameterBinding(Compress-Archive): name="CompressionLevel"; value="Optimal"
    - ParameterBinding(Compress-Archive): name="DestinationPath"; value="C:\Users\pbeesly\AppData\Roaming\Draft.Zip"
    - ParameterBinding(Compress-Archive): name="Force"; value="True"
    - ParameterBinding(Compress-Archive): name="Update"; value="False"
Check called API

- PowerShell will call .net lib, if you can hook all API then you know PowerShell's behavior.
AMSI

Ref: https://docs.microsoft.com/
Data Sources Evolution

1. Process Command Line parameter
2. Loaded DLLs
3. Windows Event Log
4. API monitoring
5. AMSI
Investigation！Not Just Detection
The key benefit for the **Red Team**

- Know more about how blue team defense
- Provide more values for organization
- Make good communicate with blue team
The key benefit for the Blue Team

- Continuous develop/validate detection
- Handle known threat first then deal with UNKNOWN
- Identify the data source you missing
- Empower the new blue team member investigation skills
Thank You!

SHANG-DE JIANG at CyCraft