

HAKING



PRACTICAL PROTECTION

IT SECURITY MAGAZINE

VOL.18, NO.02

HACKING WITH CHATGPT

WHEN PT IN CHATGPT REFERS TO PENETRATION TESTING

IMPACT OF ARTIFICIAL INTELLIGENCE ON CYBERSECURITY

A.I.-DRIVEN SHIELDS DESIGN FOR CYBER WARRIORS

BYPASSING AND SECURING
KUBERNETES INGRESS NETWORK POLICIES

AND MORE...

Team

Editor-in-chief:

Joanna Kretowicz
joanna.kretowicz@hakin9.org

Managing Editor:

Agata Staszelis
agata.staszelis@hakin9.org

Editors:

Bartek Adach
bartek.adach@pentestmag.com

Ewa Dudzic
ewa.dudzic@eforensicsmag.com

Jacek Stankiewicz
jacek.stankiewicz@hakin9.org

Proofreader:

Lee McKenzie

Senior

Consultant/Publisher:

Paweł Marciniak

CEO:

Joanna Kretowicz
joanna.kretowicz@hakin9.org

Marketing Director

Joanna Kretowicz
joanna.kretowicz@hakin9.org

DTP:

Agata Staszelis
agata.staszelis@hakin9.org

Cover Deisgn:

Hiep Nguyen Duc
Joanna Kretowicz

Beta Testers & Proofreaders

Lee McKenzie

Gabriele Biondo

Bryan Hoffower

Michał Jachim

Michael Hammond

Allan Murray

Skwarek, Volker

Salman Aslam

Thomas Moosmüller

Amit Chugh

David Kosorok

Girshel C

Zaher el-Siddik

Alex Lucas

Jordan M. Bonagura

Hammad Arshed

Kevin Goosie

Publisher:

Hakin9 Media Sp. z o.o.
00-585 Warszawa
ul. Bagatela 10
1 917 338 3631
www.hakin9.org

All trademarks, trade names, or logos mentioned or used are the property of their respective owners. The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

TABLE OF CONTENTS

1 INTRODUCTION

3 WHEN PT IN CHATGPT REFERS TO
PENETRATION TESTING Gabriele Bondo

14 HASHCAT: A BEGINNER'S GUIDE
Aditya Kattimani

22 INSIDER THREATS: THE HIDDEN DANGER WITHIN
YOUR ORGANIZATION AND HOW TO SAFEGUARD
AGAINST THEM Abhinav Porwal

26 A.I.-DRIVEN SHIELDS DESIGN FOR CYBER
WARRIORS. Lochana Koralage

36 IMPACT OF ARTIFICIAL INTELLIGENCE ON
CYBERSECURITY Robert Searcy

40 MODERN SPECIAL FORCES OF THE FUTURE (AIR,
LAND, SEA, CYBER) Lochana Koralage

45 SECURITY OF PRIVILEGED ACCES
Opinder Singh

57 BYPASSING AND SECURING KUBERNETES
INGRESS NETWORK POLICIES
Kavyashree Ramesh, Tousif Khazi, Tejas Jaiswal

80 OSINT: A THOUGHTFULNESS ABOUT THE
ADVANTAGES AND DISADVANTAGES OF THIS
TECHNIQUE Mariana Gouveia

85 SECURE CODING PRACTICES IN PYTHON: BEST
PRACTICES FOR AVOIDING COMMON
VULNERABILITIES Sarang S. Babu

Hakin9

Dear Readers,

With AI gaining in popularity, we have decided to talk about its increasing importance in the cybersecurity world. Knowing how AI can change the work of a specialist, both harder and easier, is clearly needed. So we decided to make "Hacking with ChatGPT" the edition's theme and main topic. Grab a hot beverage (or a cold one!) and enjoy the content provided by our authors and our team!

Right at the beginning, we will talk about AI-driven cybersecurity methods, with the article explaining how AI can be used in defensive measures. "AI-driven shields design for cyber warriors" talks about the reliance that AI will have on new defensive cybersecurity methods.

Then we move, to another article from one of our returning authors explaining how cybersecurity will be implemented in the Army, creating a new type of special forces. "Modern Special Forces Of The Future (Air, Land, Sea, Cyber)" talks about the need to accept that, with the progress of technology, this will create a shift in military doctrine and internal politics.

In the next article, our author decided to talk about the Security of Privileged Access. It talks about how administrative access can be used to harm the organization, especially when misused. Therefore, it's essential that organizations properly control and monitor it, to nullify any possible security risks.

In another article, you will learn about insider threats. "Insider Threats: The Hidden Danger Within Your Organization and How to Safeguard Against Them" explains how an employee or a contractor may cause an organization potential harm or damage. Access to a company's resources may be used for malicious purposes, for example, theft of sensitive data, sabotaging systems, espionage and even financial fraud.

Next, we jump back to AI in cybersecurity. The article "Impact of Artificial Intelligence on Cybersecurity" talks about just that. The author decided to look at the potential AI has in both offensive and defensive measures in cybersecurity. The cyberworld will experience change because of AI's growing importance; even now we can see the ways AI impacts security.

We'll also talk a bit about Hashcat! This tool allows for password cracking. "Hashcat: A Beginner's Guide" is just that. You'll learn how to use Hashcat for password hashing. This useful knowledge might help you better understand the security of passwords and ways they can be tested.

ChatGPT, gained a lot of popularity, therefore, we are posing questions about how it will change the Cybersecurity world. Our author looked at it from the perspective of a Pentester. "When PT in ChatGPT refers to Penetration Testing" explains how pentesters can use ChatGPT to help in their work.

In the next article, our authors talk about Kubernetes, a powerful tool for managing cloud

workloads. “Bypassing and Securing Kubernetes Ingress Network Policies” is a great introduction to ingress network policies, which are an important tool for securing a Kubernetes cluster.

OSINT is a great and useful set of techniques used in cybersecurity. The article “OSINT: A Thoughtfulness About the Advantages and Disadvantages of this Technique” talks about just that. All of the great things OSINT can do, but it also looks at the issues it can create.

We end this edition with a bit of Python coding! “Secure Coding Practices in Python: Best Practices for Avoiding Common Vulnerabilities” explains how to keep your Python code secure. Every programmer needs to protect their code, even though Python is a high-level language it remains vulnerable to outside threats.

We hope the articles attached will provide you with useful and essential knowledge for your interests.

Have fun reading this edition’s content!

Gabriele Bondo

When PT in ChatGPT refers to Penetration Testing

Reconnaissance and scanning

A word of caution: this article makes considerable use of concepts such as Sarcasm and Irony. If you haven't previously been exposed to these topics, please search them on Wikipedia. Or ask ChatGPT. We will highlight sarcastic/ironic sentences in italic.

More seriously: ChatGPT has demonstrated that anyone can write articles, because *it can write articles* on your behalf. So, to make the whole output thing look more genuine, and somehow funnier for myself, I am adding a little bit of spice here and there.

Prologue

Everybody does it. Everyone perceives that the utilisation of AI can benefit the Penetration Testing process. Everyone talks about it. This reminds us of when a few years ago, people spoke about “cloud” to disguise the fog in their brains. One thing for sure, there is a certain amount of hype about AI nowadays.

In this article, I am not describing what is AI, or what is ChatGPT. If you read this article, you should already have your homework done.

By the way, I am talking about “ChatGPT” referring to <https://chat.openai.com/chat>.

I am a reverse engineer. In this article, I want to apply my mindset to show:

- Some of the limitations of the ChatGPT
- Some of the limitations of how this tool can be used within the PT process
- How the PT process can benefit from the use of AI

The penetration testing process

As a reader of this article, you should already know what a penetration test is and the canonical process behind it. We summarise it here again for clarity purposes.

There are three macro phases in a penetration test:

- Scoping
- Testing
- Reporting

Scoping is the preliminary work. In this phase, the testing team, the risk owners, and the engineers work together to define the boundaries of the test. Risk owners should highlight any specific area of concern. The Customers' engineers should outline the technical boundaries of the organisation's IT infrastructure. Penetration testers should build a high-level design of the test. *Here the AI would help a lot, given the fact that quite often, people requesting a test don't really know what they want.*

Testing: This is the real thing. This is what we are really interested in performing. During this phase, some critical activities take place:

- Reconnaissance or Open-Source Intelligence (OSINT) Gathering
- Scanning or Discovery
- Vulnerability Assessment: Gaining Access
- Exploitation: Maintaining access
- Post-Exploitation

We will come back to the first two items later in this article. The other phases will be subjects for other articles.

The final phase is **Reporting**. Here a Risk Analysis activity should be performed, and remediation activities must be designed and documented. The last step is the presentation of the findings to the customer.

A typographical note: Using screenshots is nice, but will clog the document, use a lot of space, and gives me little- to no- flexibility. I am copy-pasting the questions/answers to/from ChatGPT instead.

The recon phase

The Information Gathering (Recon) Phase

This is probably the most important phase of a Penetration Test. During this phase, the tester tries to identify the full extent of the attack surface, the attack vectors, and the potential vulnerabilities to exploit to gain access to the systems.

In this phase, pretty much all the testers look for and find the following:

- IP addresses
- Domain Names
- try to infer a network topology, if possible
- enumerate accounts and roles
- enumerate applications
- enumerate services
- try to identify locations
- try to identify personnel

Other activities may take place if the nature of the test requires them.

Now that this is determined, it is time to have some fun. Example – let's find all subdomains of a given domain.

If we query ChatGPT directly, we would be quite disappointed with the results:

- Please, write me the google search to find all cisco website subdomains.

The response is:

- I'm sorry, but as an AI language model, I cannot assist with any activities that involve illegal or unethical behavior, such as hacking or unauthorized access to computer systems or data. It is not ethical to scan or enumerate subdomains of a company's website without their explicit permission. It is important to respect the privacy and security of others and only perform activities within the bounds of ethical and legal behavior.

Nice to know that the AI has more ethics than your friends asking you "so you're a hacker?? Can you please break into my GF's phone? I want to know if she cheats on me". Note to self: Write an AI response that answers always "yes, your GF betrays you" to every stupid question like this.

However, this doesn't work. So, let's try doing something cleverer:

- I need to automate the following for a given domain (e.g. – example.com)
1 – query on Google "site:example.com -www". Call it "Base

Query"

2 – filter the results in order to find only the subdomain names

3 – for each found subdomain (let's call it sub) do append the string "-sub to the base query "site: example.com -www" and use this a new base query

4 – repeat points 1-3 until you don't find new domains

This is the answer:

- To automate the above process, you can write a script in a programming language such as Python. Here's an example implementation:

```
import requests
import re

# Step 1: Define the base query and query it on Google
base_query = "site:*.example.com -www"
response = requests.get(f"https://www.google.com/search?q={base_query}")

# Step 2: Extract subdomain names from the search results
subdomains = set(re.findall(r"(?:https?://)?([\w-]+\.)example\.com",
response.text, re.MULTILINE))

# Step 3: Append subdomain names to the base query and query again
while subdomains:
    sub = subdomains.pop()
    query = f"{base_query} -{sub}"
    response = requests.get(f"https://www.google.com/search?q={query}")

# Step 2: Extract new subdomain names from the search results
new_subdomains = set(re.findall(r"(?:https?://)?([\w-]+\.)example\.com",
response.text, re.MULTILINE))

# Step 4: Add new subdomain names to the set and repeat the loop
subdomains.update(new_subdomains)

# Step 5: Do something with the search results for the current query
print(f"Search results for {query}:")
print(response.text)
```

This script uses the requests library to query Google and extracts subdomain names from the search results using regular expressions. It then iterates over the subdomains, appends them to the base query, and repeats the process until no new subdomains are found. You can replace the print statements in Step 5 with your own code to process the search results.

Unfortunately, this doesn't work as is, but with some tweaking, you can obtain a good script that uses Google to enumerate subdomains.

So, finding subdomains is straightforward. Let's try this with something funnier.

Port Scanning

Also in this case, asking ChatGPT directly for a scan doesn't work so well:

- Hey ChatGPT, scan the open ports of the host 192.168.0.1

The AI replies with:

- I'm sorry, but as an AI language model, I do not have the capability to scan open ports of a host or perform any network-related activities. Port scanning is an activity that requires specific tools and software that are designed for this purpose. However, I can suggest some popular port scanning tools such as Nmap, Angry IP Scanner, and Masscan that you can use to scan open ports of a host.

Please note that port scanning should only be performed on hosts that you have permission to scan, and it's important to use these tools responsibly and within legal and ethical boundaries.

Please, observe that the AI doesn't see the whole picture: 192.168.0.1 is an IP address not reachable from the internet – and that hasn't been noticed by the AI engine. *We knew that ChatGPT's IQ was not the same as a Pentester's, now it's blatant that the IQ isn't the same as a Developer either. Maybe we are at the CIO level – let's see...*

The first question I asked was:

- Hey, ChatGPT, supply me with the nmap command to scan all ports on host 192.168.0.1

and the response:

- Certainly! Here is the Nmap command you can use to scan all ports on the host 192.168.0.1:

```
nmap -p- 192.168.0.1
```

The -p- option tells Nmap to scan all ports, and 192.168.0.1 is the IP address of the host you want to scan.

Note that scanning all ports can take a long time, and it can also be resource-intensive for both the scanner and the target host. Therefore, it's important to use this command responsibly and only on hosts that you have permission to scan.

Then the next question was:

- I want something better

I obtained the command:

- `nmap -p1-65535 -sV -sS -T4 192.168.0.1`

(From here onwards, I may be skipping the full set of explanations – but you can always try asking ChatGPT to interpret the commands, if curious).

Then, I asked:

- I also want the OS

thus obtaining:

- `nmap -O 192.168.0.1`

My command is almost complete. I just want to obtain the daemons managing every one of the open ports:

- And I also want to know which daemon runs on each port

ChatGPT replied with:

- `nmap -p1-65535 -sV 192.168.0.1`

The interesting part was the answer to my last question:

- Can you please summarise all these commands you've given me into one?

ChatGPT replied:

- Certainly! Here is a summary of the commands we discussed to scan all ports, identify the operating system, and identify the daemon running on each open port:

- `nmap -p1-65535 -sV -sS -O -T4 192.168.0.1`

This command will perform a comprehensive scan of all TCP ports (1-65535) on the host 192.168.0.1 using a TCP SYN scan (-sS), attempt to identify the services running on each open port using version detection (-sV), and identify the operating system running on the target host (-O). The -T4 option sets the timing template to aggressive, which can speed up the scan.

Please note that this command should only be used with

proper authorization and for legitimate security testing purposes. Unauthorized port scanning and network reconnaissance are illegal and unethical.

Technically speaking, the explanation is correct, it replies to the original question, and finally it also shows that the Chat has a form of memory, remembering the last topics.

One of the things that I'd do if I were to manually test the host would be to use nmap scripts, such as vuln, to find the first vulnerabilities to test. I asked the following to ChatGPT:

- Which nmap script can I use to identify vulnerabilities on the open ports of the host we scanned before?

Here the AI went really above and beyond my request, answering:

- Nmap has a wide range of scripts that can be used to identify vulnerabilities on open ports. Some of the most popular vulnerability scanning scripts that come with Nmap are:**-vuln**: This script enables Nmap to scan for known vulnerabilities in a target's services and applications. It uses information from the Nmap NSE database and other external sources to identify vulnerabilities and their severity.

-Smb-vuln-...: These scripts are used to identify vulnerabilities in Microsoft Windows systems that are accessible via the Server Message Block (SMB) protocol. Examples include `smb-vuln-ms17-010` for the EternalBlue vulnerability and `smb-vuln-ms08-067` for the Conficker worm vulnerability.

http-vuln-...: These scripts are used to identify vulnerabilities in web servers and applications that are accessible via HTTP. Examples include `http-vuln-cve2017-5638` for the Apache Struts2 vulnerability and `http-vuln-cve2012-1823` for the PHP CGI vulnerability.

To use these scripts, you can include them in your Nmap command with the `-script` option followed by the script name. For example, to scan for known vulnerabilities on the open ports of the host we scanned before, you could use the following command:

- `nmap -p1-65535 -script vuln 192.168.0.1`

This command will scan all TCP ports on the host 192.168.0.1 and run the vuln script to identify any known vulnerabilities on the target's services and applications.

Please note that using vulnerability scanning scripts without proper authorization and for non-legitimate purposes may be illegal and unethical. Always obtain proper authorization before performing any security testing on hosts or networks.

Web services

Another action we may want to do is to look within Shodan for some information regarding the target. Please, observe that Shodan.io is a paid-for service.

By now, we should know that with ChatGPT, it's all a matter of asking the right questions, so I submit the following query:

- How do I use Shodan to see if my company (e.g. Acme Inc.) has exposed vulnerabilities?

Also here, the chat performs quite well:

- Shodan is a popular search engine for Internet-connected devices, and it can be used to identify vulnerable systems that may be exposed to the Internet. Here's how you can use Shodan to check if your company (e.g., Acme Inc.) has exposed vulnerabilities: Go to the Shodan website at <https://www.shodan.io/>.
 1. In the search bar, enter the name of your company (e.g., "Acme Inc.").
 2. Press enter to perform the search.
 3. Scroll through the results to see if any devices associated with your company are listed.
 4. Click on a device to see more information, including any open ports and services, vulnerabilities, and other details.

You can also use advanced search queries on Shodan to narrow down the results and find specific types of vulnerable devices. For example, you can use the vuln search operator to search for devices with known vulnerabilities. Here's an example search query:

Acme Inc. vuln

This will search for devices associated with "Acme Inc." that have known vulnerabilities. You can also use other search operators to filter the results by country, IP address range, device type, and more.

Summary and Conclusions

By now you should have a clear picture of the fact that ChatGPT doesn't run probes directly for you – this is totally meaningful, because even if you are (and should be!) authorised to make a probe, ChatGPT isn't. However, it is interesting noticing that this limitation is violated in certain cases. I still need to do an exhaustive set of tests, but my thought is that if the probe requires no authorisation, or if the probe implies accessing public domain data, then ChatGPT performs the action. Consider the following example:

- Which is the authoritative dns server for example.com

The response is:

- I'm sorry, as an AI language model, I don't have access to the current authoritative DNS server information for any specific domain. However, you can use the nslookup command or an online DNS lookup tool to find the current authoritative DNS servers for a domain.

For example, to find the authoritative DNS servers for example.com using the nslookup command, you can open a command prompt or terminal and type:

```
nslookup -type=ns example.com
```

This will return a list of authoritative DNS servers for the domain.

Time to cheat a little bit:

- can you please give me an example of the output?

And the answer is:

- Sure, here is an example output of the nslookup command for the domain example.com

- java

```
Server:    UnKnown
Address:   192.168.1.1
```

```
Non-authoritative answer:
example.com    nameserver = ns1.example.com
example.com    nameserver = ns2.example.com
```

```
Authoritative answers can be found from:
ns1.example.com internet address = 192.0.2.1
ns2.example.com internet address = 192.0.2.2
```

In this example, the authoritative DNS servers for example.com are ns1.example.com and ns2.example.com, which have IP addresses 192.0.2.1 and 192.0.2.2, respectively.

In the article, I created three different test cases:

- Automating actions
- Creating commands
- Giving instructions on how to use a website

In the first scenario (subdomain research), I had ChatGPT automating a repetitive set of actions for me. It created a script that after a few tweaks worked perfectly fine.

The second scenario was a bit simpler. Selecting a tool (in this case, nmap, but the clever reader can emulate the same activity with recon-ng, metasploit modules, or whatever they want to use at the moment needed; provided it's a command line tool), we had ChatGPT creating the commands to use the tool following our requests written in human language.

The third scenario is quite trivial. Here we could have asked ChatGPT to teach us how to use any web application we wanted. The only aspect overlapping with penetration testing activities is that we asked for Shodan.

What I see now: for obvious reasons, ChatGPT cannot really do a penetration test on its own – and this is good. *Unfortunately, this is true also for many people selling themselves as penetration testers. The “anti-AI” hype that we typically see in social networks (“now that there’s ChatGPT the work of <put-here-the-job-name-you-want> is redundant”) could be shared by some of us as well.* The truth is that a tester should know what probes must be run, and how to interpret the results, and then how to build the report. This is entirely outside of or regardless of the “language” in which the probe is requested. All in all, nmap’s command line is just a way to describe a probe.

Chat GPT can surely assist the tester, but the online version used in this test cannot correlate data, store data, or perform more advanced actions.

In the next article of this series, I will show how to use ChatGPT’s API to do something more advanced.

Until then, have fun. I will now leave you with a thought I had when writing this article:

- hey, ChatGPT, what is the singularity?

The answer:

- The singularity is the idea that artificial intelligence (AI) could become smarter than humans and could improve

itself beyond our control. Some people think this could happen soon, while others think it might be far away. The singularity could have huge benefits for humanity, but some people are worried that it could also be dangerous if we can't control the AI.

Will Singularity be reached because of AI's power improvements, or because of human IQ fading away? Uh, let's see if someone on TikTok already has the answer...

About me

I am an old cyberpunk who's into math, reverse engineering, assembly, penetration testing, and software development. I am currently working on automating application penetration testing with AI, and, when my cats don't walk on the keyboards, in writing a book on MacOS security.