UNDER
DEFENSE
CyberSecurity Solutions
Protecting your business

Fuzzing Project

Grey Box Fuzzing

Service Title	Fuzz Testing
Client Overview	Our client is the world's leading provider of innovative physical security products and solutions such as CCTV cameras and other.
Client Challenge	Prepare module for multi-protocol fuzzer tool which will cover all processes related to CCTV cameras work.
Scope	The scope of this testing is bounded by specified protocols realisations and network device communications.
Workflow	UD figured out with all specifications related to CCTV cameras work and prepare a module for fuzz testing.
Key Benefits	Our client can now perform comprehensive security testing and find the security weaknesses before deployment and before hackers do that.
Results	The results of our project is a comprehensive module which allow to conduct all security testing process via rich and stylish GUI.

UNDER
DEFENSE
CyberSecurity Solutions
Protecting your business

What is Fuzz Testing?

Fuzzing is an excellent technique for locating vulnerabilities in software. The basic premise is to deliver intentionally malformed input to target software and detect failure. A complete fuzzer has the next steps:

Market Investigation

The CCTV market is gaining huge popularity across the world due to rising concerns for security and safety. This has resulted into an increased demand for technically advanced surveillance system, thereby, creating huge growth opportunities for CCTV manufacturers. Different market research such as here shows that at the next few years a Global CCTV Market will be increasing by **$ 3 billions** per year.

$3,000,000,000

annual growth of CCTV market *

Client Challenge

The client asked to create a module for multi protocol fuzzer tool which covers all processes related to CCTV cameras work, and trying to find programme issues in million different places.

FUZZING MODULE

WS-username token	0%	10%
HTTP Digest Auth	0%	30%
RTSP Stream	0%	0%

HTTP Methods	0%	100%
Onvif device service	0%	40%
HTTP Headers	0%	50%
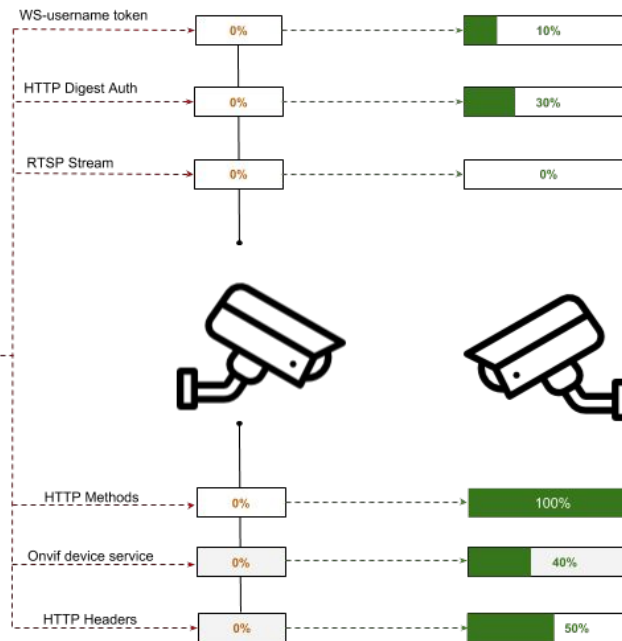
UNKNOWN, %

CCTV Cam Health Status

40%

Overall Health Status

UNDER DEFENSE
CyberSecurity Solutions
Protecting your business
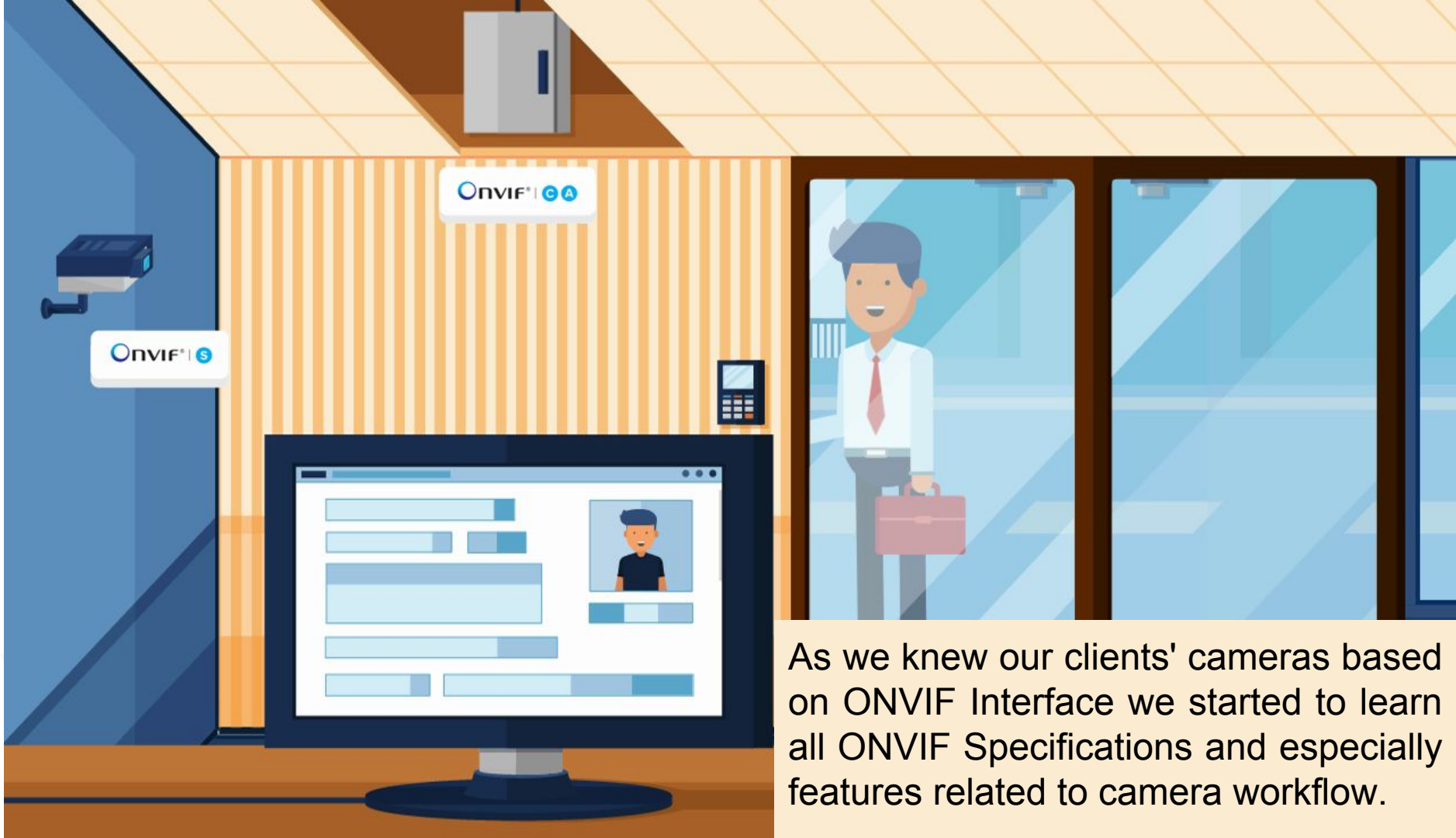
Fuzzing Scopes
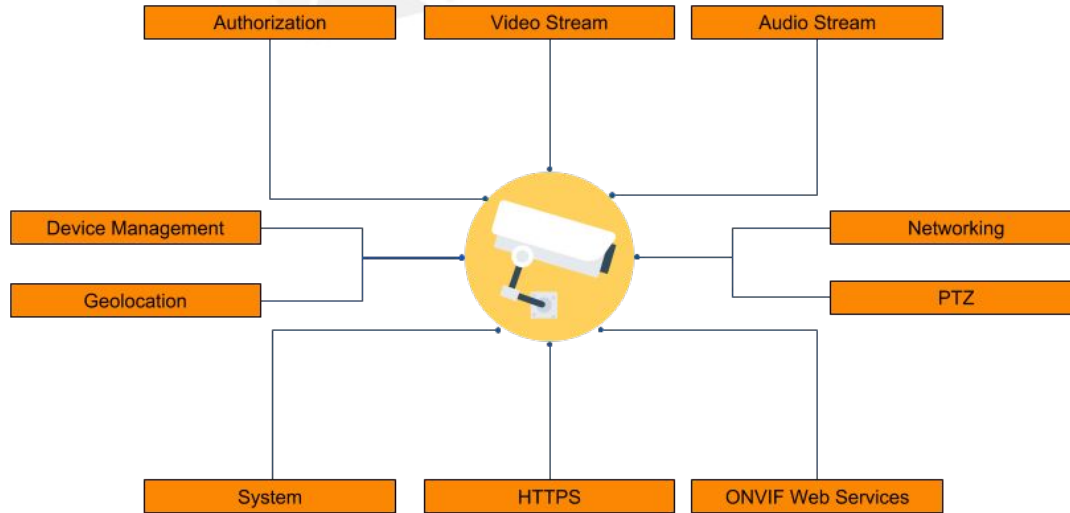
The scope of this project includes fuzz testing which based on ONVIF Specifications. An ONVIF describes six profiles each of which has a fixed set of features. The product of our client is conformant with ONVIF Specifications so that we defined a certain area for all attack vectors which we need to fuzz.

As we knew our clients' cameras based on ONVIF Interface we started to learn all ONVIF Specifications and especially features related to camera workflow.

```
POST /onvif/device_service HTTP/1.1
Host: 192.168.100.31
Content-Length: 251
Authorization: Basic YWRtaW46YWRtaW4=
Postman-Token: 8064d56a-1c83-a479-5371-5a5fd24dbe93
Cache-Control: no-cache
Origin: chrome-extension://fhbjgbiflinjbdggehcddcbncdddomop
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36
Content-Type: text/xml
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,uk-UA;q=0.8,uk;q=0.7,ru;q=0.6
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:tds="http://www.onvif.org/
ver10/device/wsdl">
 <SOAP-ENV:Body>
    <tds:GetSystemDateAndTime/>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The ONVIF conformant products use web services which handling SOAP requests for communication between devices and clients. This is described in ONVIF Network Interface Specification. So that our purpose is a discovering all possible requests on the attack surface and create a module which will conduct fuzz testing based on this requests.

UNDER
DEFENSE
CyberSecurity Solutions
Protecting your business

8

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:tds="http://www.onvif.org/ver10/device/wsdl">
 <SOAP-ENV:Body>
    <tds:GetSystemDateAndTime/>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>


<SC Name="Data">
    <SE Name="HTTP Headers">
        <S Name="Requst Line">
            <C Name="Method" ASCIIValue="POST" />
            <C Name="Space" ASCIIValue=" " />
            <C Name="URI" ASCIIValue="/onvif/device_service" />
            <C Name="Space" ASCIIValue=" " />
            <C Name="Protocol" ASCIIValue="HTTP" />
            <C Name="Forward slash" ASCIIValue="/" />
            <C Name="HTTP Major Version" ASCIIValue="1" />
            <C Name="Dot" ASCIIValue="." />
            <C Name="HTTP Minior Version" ASCIIValue="1" />
        </S>
        <S Name="HTTP Host Header">
            <C Name="HTTP Host Header Name" ASCIIValue="Host" />
            <C Name="Colon" ASCIIValue=":" />
            <C Name="Space" ASCIIValue=" " />
            <C Name="Address" ASCIIValue="192.168.100.31" />
            <C Name="CRLF" Value="0x0D,0x0A" />
        </S>
        <S Name="Content Length Header" />
            <C Name="CL" ASCIIValue="Content-Length" />
            <C Name="Colon" ASCIIValue=":" />
            <C Name="Space" ASCIIValue=" " />
            <C Name="CL Value" ASCIIValue="251" />
            <C Name="CRLF" Value="0x0D,0x0A" />
        </S>
        <S Name="Auth Header" />
            <C Name="Auth Header Name" ASCIIValue="Authorization" />
            <C Name="Colon" ASCIIValue=":" />
            <C Name="Space" ASCIIValue=" " />
            <C Name="Auth Type" ASCIIValue="Basic" />
            <C Name="Space" ASCIIValue=" " />
            <C Name="Auth Value" ASCIIValue="YWRtaW46YWRtaW4=" />
            <C Name="CRLF" Value="0x0D,0x0A" />
        </S>
        <S Name="Postman Token Header">
```

The fuzzing module are created using an special XML syntax. And this module defines both ways how it looks like, acts and what type of data it sends as well as how it should generate the fuzzed data.

We predefined a list of payloads based on attack area and integrated it's on our requests.

UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

Test Progress

(View Module Browser)
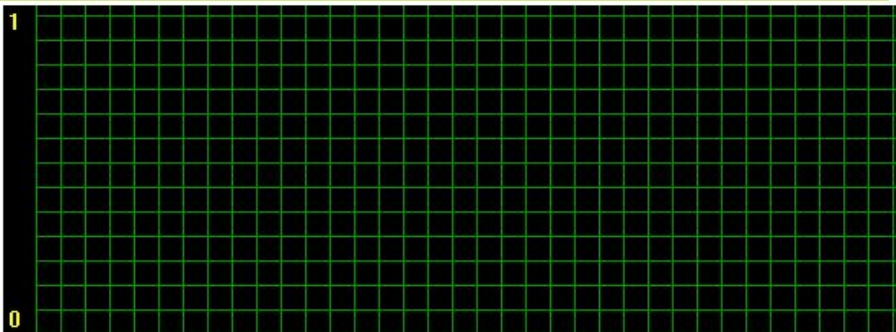
* Yellow line - Average   * Green line - Momentary

Saturation Rate Threshold:
Momentary: 0   Avg.: 0.00   Max: N/A   Min: 0

Preview

Addr: 0000 Hex: 50 Dec: 80 Bin: 01010000 Ascii: P Length: 355 (0x163)

```
0000  50 4F 53 54 20 2F 6F 6E 76 69 66 2F 64 65 76 69   POST /onvif/devi
0010  63 65 5F 73 65 72 76 69 63 65 20 48 54 54 50 2F   ce_service HTTP/
0020  31 2E 31 0D 0A 48 6F 73 74 3A 20 31 39 32 2E 31   1.1  Host: 192.1
0030  36 38 2E 31 30 30 2E 33 31 0D 0A 43 6F 6E 74 65   68.100.31  Conte
0040  6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61   nt-Type: applica
0050  74 69 6F 6E 2F 73 6F 61 70 2B 78 6D 6C 3B 20 63   tion/soap+xml; c
0060  68 61 72 73 65 74 3D 75 74 66 2D 38 0D 0A 43 6F   harset=utf-8  Co
0070  6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 32 32   ntent-Length: 22
0080  32 0D 0A 0D 0A 3C 3F 78 6D 6C 20 76 65 72 73 69   2    <?xml versi
0090  6F 6E 3D 22 33 31 2E 30 22 20 65 6E 63 6F 64 69   on="31.0" encodi
00A0  6E 67 3D 22 55 54 46 2D 38 22 3F 3E 0D 0A 3C 53   ng="UTF-8"?>  <S
00B0  4F 41 50 2D 45 4E 56 3A 45 6E 76 65 6C 6F 70 65   OAP-ENV:Envelope
00C0  20 78 6D 6C 6E 73 3A 53 4F 41 50 2D 45 4E 56 3D    xmlns:SOAP-ENV=
```

PREVIEW   EXPORT

Attack Vector: M0:P0:B0.BT0:B0.BT0:B0.BT0:B0.BT0:B0.BT0:B0.L0:B0.BT0:B0.BT0:B0.BT0:B0.BT0:B0.BT0:B0.BT0:

Project Settings

Name	Value
Version:	7.3.20 (build 7065)
Project Name:	projecttestSOAP
Thread Count:	1
Remote Monitor IP Address / Hostname:	192.168.100.31

Module Settings

Module name:	SOAP over HTTP
Fuzzing conditioned elements:	Yes
Generator type:	Binary
Increment order:	Normal
Overflow elements once:	No
Scale type:	Base2+/-1
Saturation Rate Threshold:	100
CPU Based Saturation Rate Threshold:	no
Batch Mode:	yes
Report Connectivity Issues as Exceptions:	no
Minion Enabled:	no
Minion Host and Port:	unset (6980)
Module Type:	Network Client

Environment Settings

Name	Value	Ty
HTTP Host Value	<Hostname to Test>	No
Name Space Method	CreateProfile	No
Name Space Property Name	Token	No
Name Space Property Value		No
Name Space URL	http://www.w3.org/2003/05/soap-envelope	No
Remote Hostname	192.168.100.31	No
Remote Port	80	No

The latest step is a monitoring and report which shows all anomalies found during module operation.

UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

10

Workflow

Creating Fuzzing MVP

1

At this step, we prepared a module for a generic fuzzing framework which includes a general algorithm for fuzzing purposes.

Integrating Monitoring Process

3

Integration a tool that looks for anomalies and vulnerabilities in the system that can lead to critical consequences.

Final results

5

In the results of our project, we provide a possibility to test CCTV ONVIF conformant products in a certification test or as part of the development lifecycle.

Enhancing Functionality

2

At the second step, we enhanced module algorithm by functions which realize client and device communications and covers all features related to CCTV workflow.

Report Generating

4

After successful fuzzing process, we got a report which shows all attack vectors, malformed data and vulnerabilities which was found on the target.

UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

Key Benefits

1. One platform fo make all security fuzz testing.
2. Fuzz tests cover all device functionality which can be affected by multiple vulnerabilities.
3. Great monitoring and reports about each found anomaly.
4. Possibility to make product secure and defend it against attackers.
5. Increases a chance to receive security certifications which shows your reliability and competitive on the market.

1

2

3

4

5

UNDER
DEFENSE
CyberSecurity Solutions
Protecting your business
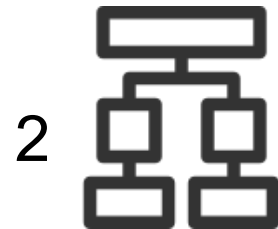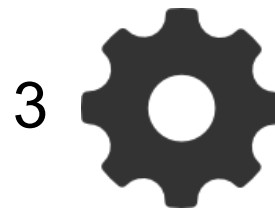
Thank you!

Call us now at +1 929 999 5101