Case Study

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:

UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

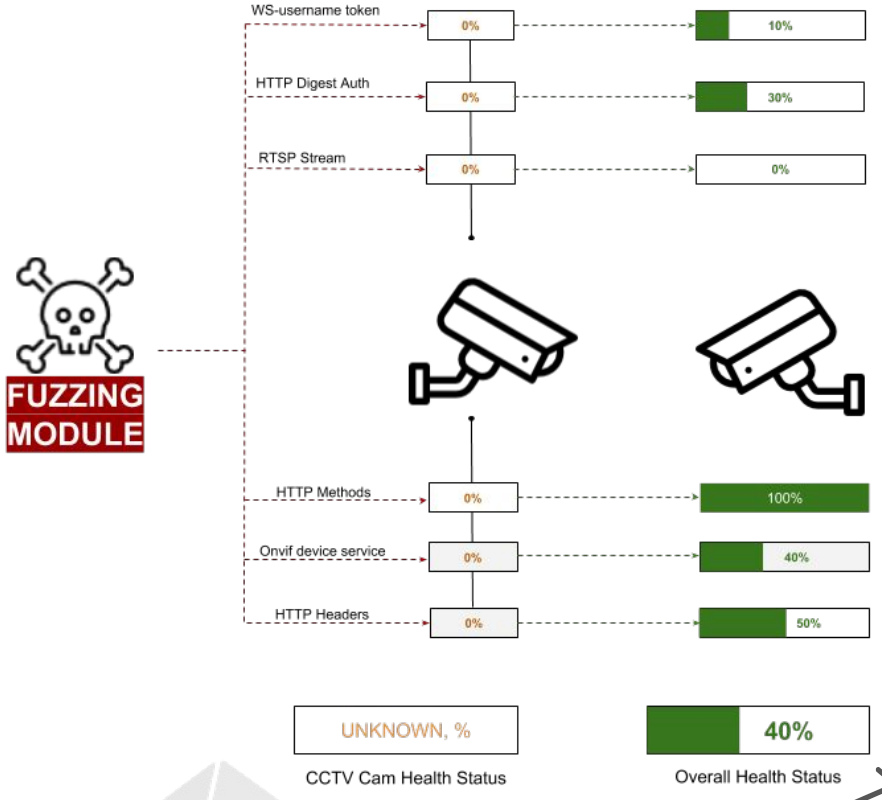
Market Investigation

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.

* https://www.researchandmarkets.com/research/bjjn6d/global_cctv?w=12

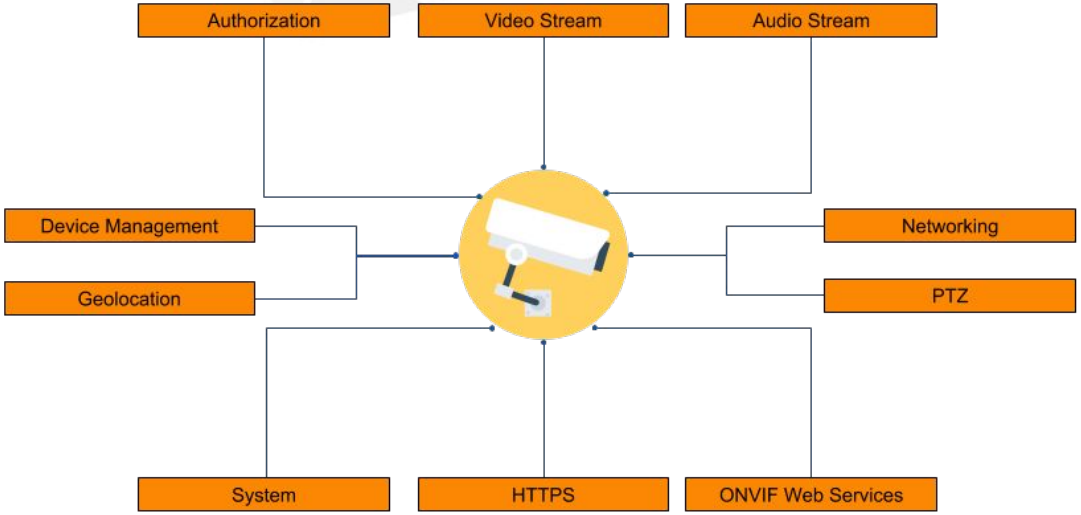
Client Challenge

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.



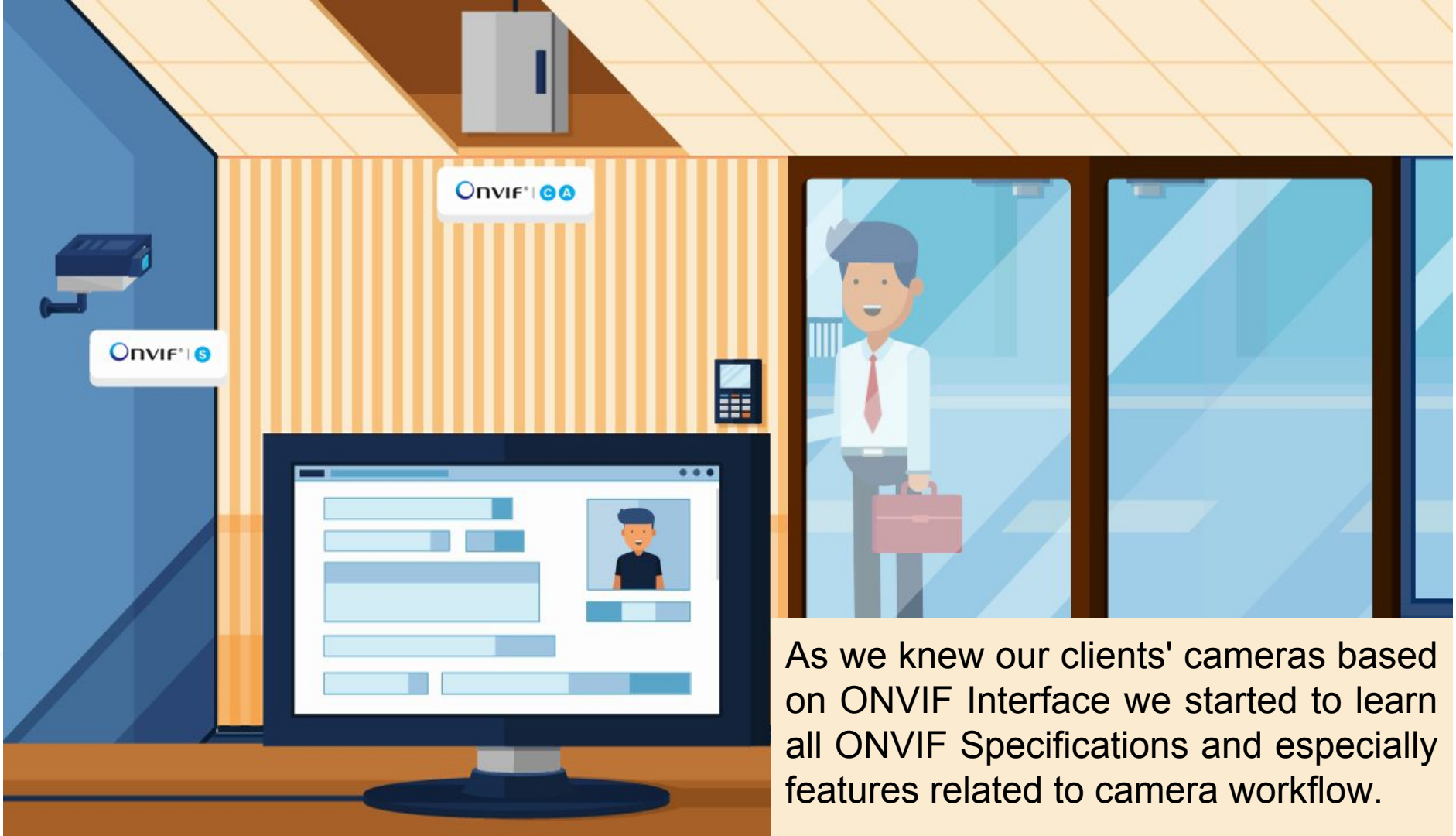
UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

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.

UNDER DEFENSE
CyberSecurity Solutions
Protecting your business



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.

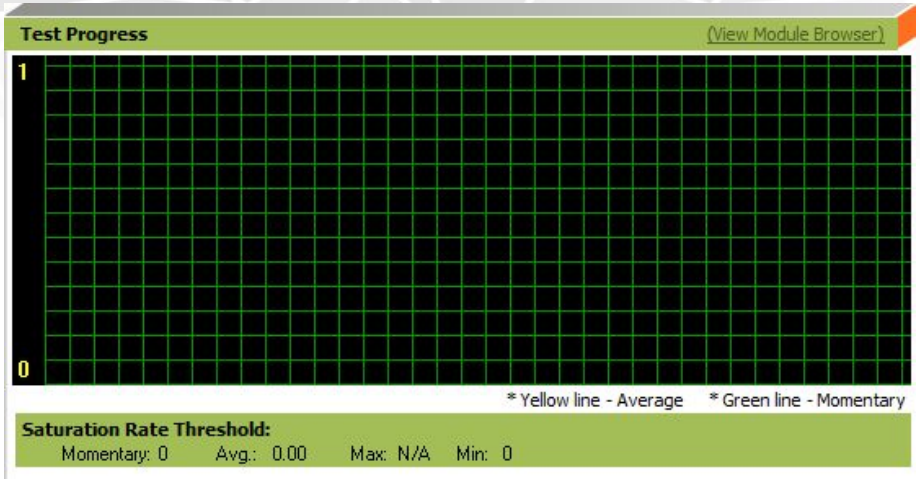
UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

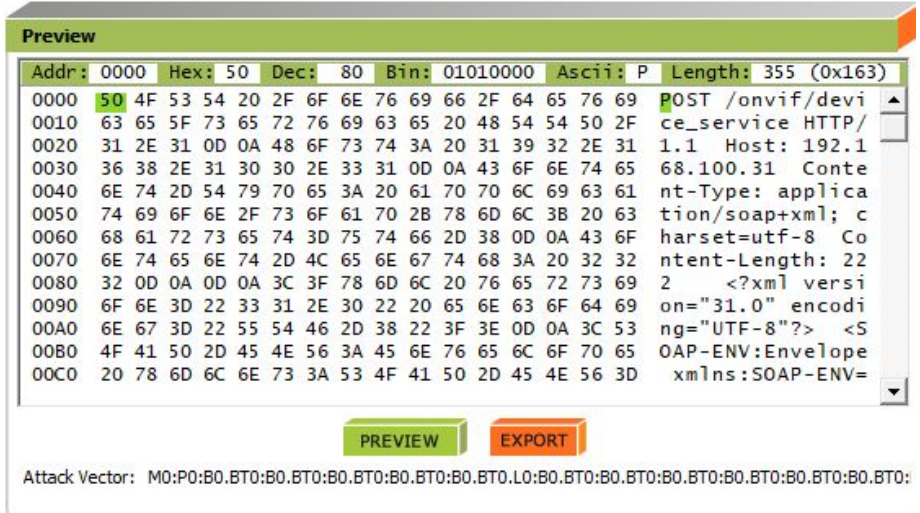
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:tds="http://www.onvif.org/
ver10/device/wsdl">
 <SOAP-ENV:Body>
    <tds:GetSystemDateAndTime/>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>


<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





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>	Nc
Name Space Method	CreateProfile	Nc
Name Space Property Name	Token	Nc
Name Space Property Value		Nc
Name Space URL	http://www.w3.org/2003/05/soap-envelope	Nc
Remote Hostname	192.168.100.31	Nc
Remote Port	80	Nc

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

UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

Workflow

1 Creating Fuzzing MVP

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

2 Enhancing Functionality

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

3 Integrating Monitoring Process

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

4 Report Generating

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

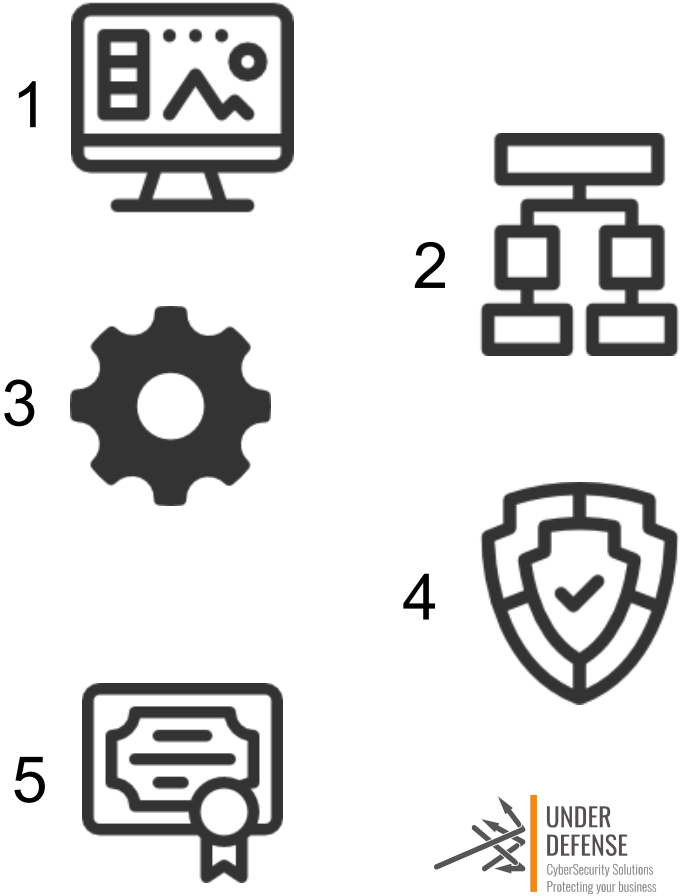
5 Final results

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.

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.



UNDER DEFENSE
CyberSecurity Solutions
Protecting your business

Thank you!

Version 2 | 二版
www.version-2.com
Hong Kong | Taiwan | Singapore | Macau | Mainland China

Hong Kong & Macau
Tel : (852) 2893 8860
Email : sales@version-2.com.hk

Taiwan
Tel : (886) 02 7722 6899
Email : sales@version-2.com.tw

Singapore, Malaysia & SEA
Tel : (65) 6296 4268
Email : sales@version-2.com.sg