Defense from Covert Channel Cyber-attack over video stream payload

Y. Segal, E. Hadas, R. Birman, and O. Hadar

Communication Systems Engineering Department,
Ben Gurion University of the Negev (BGU),
Beer-Sheva, 84105,
Israel[1]

***Abstract*— Attackers find the multimedia world in general and video streaming in particular, an attractive backdoor for cyber-attacks. More and more people consume (share, post and download) on-line Multimedia data, this aspect makes targeting those media types very attractive to hackers. Multimedia covert channels provide reasonable bandwidth and long lasting transmission streams, suitable for planting malicious information and therefore used as an exploit alternative. Most attack algorithms via video, focus on planting malicious information in video headers. Such techniques are easy to detect and provide limited transmission bandwidth, therefore increasing attackers' motivation to use video payload instead. In this research we propose a method to protect against attacks that use video payload for transferring confidential data using a covert channel. The payload is the part of the transmitted video data that carries the actual picture content viewed by end-users, making it more challenging to protect against, since the content is less structured and can carry any arbitrary image data content.**

In this paper, we will demonstrate attacks that take advantage of compressed domain video payload and propose a set of defense techniques for improving defense efficiency.

Keywords*— *Exploit; Invisible Covert Channel; Steganography; Watermarking; Cyber; Stego objects; Discrete Cosine Transform; DCT; Motion Vectors

I. INTRODUCTION

Video steganography is the process of secretly inserting and concealing data within videos. Steganography has been helpful in protecting media copyrights (via digital watermarks). On the extreme end, sophisticated users have used steganography as means of communication, transmitting hidden messages without anyone, but the intended recipient/s, being aware of it. Lately, newspaper reports have indicated that some users are using malicious software to break into smartphones, computers and even internet-connected televisions.

Multiple techniques have been reported for steganography and watermarking. An overview of digital image steganography is presented in [2]. In [3] basic building blocks for steganography in compressed video were examined: the embedding operation and the choice of embedding alternatives.

It is shown in [4] that Facebook Cover Photos can effectively hide information using Discrete Cosine Transform (DCT) coefficient embedding algorithms. Watermarking solves the challenge of illegal video distribution and manipulation. Watermark's robustness is critical for avoiding attackers' watermark disruption. Some methodologies were developed in [5] for compressing the robustness of different watermarking techniques. The watermarking algorithm presented in [6] embeds the watermark into the video by adjusting intermediate frequency coefficients.

An innovative approach for cyber-attack/defense applying a Smart threshold and Anomaly Correction to compressed domain DCT coefficients is described in [7]. In this paper we focus on manipulations of compressed domain Error estimation of Motion Vectors.

Video compression protocols, such as H.264, for example, divide video frames to Macro Blocks (MB) and estimate their movement between frames (represented by Estimated Motion Vectors, or EVMs). In order to minimize the number of bits required to represent transmitted EMVs, the algorithm assumes that neighboring MBs statistically move in similar vector values (size and direction). Therefore, it is possible to transmit only the EMVs delta values compared to neighboring MBs. In order to find the EMV delta value, a Median of all neighboring MBs EMV is calculated and only the delta is sent (error estimation of the Motion Vector).

The Cyber-attack algorithm takes advantage of lack of sensitivity of movie viewers to small deviations of Macro-Block (object) movements from their original path. Viewers are not likely to notice minor noise around moving MBs. Moreover, since the viewer does not know the accurate real position of MBs in the original video movie, they are not likely to notice minor displacement changes that affect MBs position accuracy.

II. OBJECTIVE

The objective is to fight against those growing multimedia threats, using a series of algorithms that are designed to prevent hackers from exploiting pictures and videos to gain access to personal and confidential information.

The cyber-attack algorithm is performed in the Compressed Domain. While most payload attacks manipulate the Pixels Domain, the Compressed Domain yields better performance results with less computational power, due to the fact that it is applied on-the-fly without decoding the stream. The algorithm is based on identifying MBs with relatively small displacement movement compared to other neighboring MBs, and using their displacement content as place holders for malicious data. Common video streams contain many such MBs, thus

[1] This work was supported by the Israel National Cyber Bureau.

providing an effective exploit covert channel with reasonable bandwidth potential (kbps).

Our objective is to devise a new technique that can potentially eliminate some types of cyber-attacks initiated via video, audio or pictures from the internet.

III. Attack/ Defense algorithm

Cyber protection algorithms have two basic models: Detection and Prevention. Detection is an alerting algorithm that typically uses signature analysis or statistical anomaly detection methods. It has the advantage of being attack specific, but may not be able to generalize. The generalization gap is overcome by incorporating some automatic adaptation in the detection process, or implementing some learning cycles, which might consider an attack as normal data. Prevention is a process that prevents malicious data from penetrating the site or the system. The Prevention process operates on a regular basis regardless of the existence or non-existence of attack, therefore, providing more general protection compared to the Detection process.

In this research work, we are exploring a real-time Prevention algorithm for H.264 video streams. It is part of a more General Prevention Research (GPR) against attacks that use the video or audio stream payload as a malicious data container.

Payload manipulations produce some artifacts that can be described as noise addition to original video stream images. Modern video coding techniques employ lossy coding schemes, which often create compression artifacts that may lead to degradation of perceived video quality. Payload attack takes advantage of naturally introduced compression artifacts and assumes that the user will not be able to distinguish between compression artifacts and malicious data of covert channel artifacts.

A. *Attack perspective*

To be able to prevent attack via video, it is necessary to analyze and understand the attacker point of view.

Video based Cyber-attacks can be divided into two stages: first, the planting of hostile malware which will perform offensive actions such as: taking control of the device, deleting information, denial of service and so on.

The second stage is establishing of a hidden communication channel (covert channel), capable of communicating with the malicious software that was preinstalled and sending to it remote operation commands, such as timing the attack and determining the type of attack. In advanced attacks, the covert channel can be used to manage a rolling event, whereas the attack develops according to victim’s responses.

The paper is focused on offensive prevention of the second cyber-attack stage (the covert channel), assuming that the hostile software already exists on the victim system. The first cyber-attack stage is out of the scope of this paper.

Attackers objective is maximizing covert channel bandwidth, thus maximize the amount of malicious data delivered in the stream payload, while minimizing the noise level.

There are two types of such video attacks – Online and Offline. Offline attacks are based on recorded movies. The attackers have access to, or have some movies that they promote. This situation provides attackers with all the time that they need to plant malicious data in the video.

Online attacks are much more complicated because attacks are based on intervening between the content streaming server and the user (man-in-the-middle attack). The online interference needs to guaranty very low latency. Brute-force payload manipulation requires online video transcoding process (decoding and encoding). The transcoding process consumes processing time and increases the latency. Therefore, online attacks will usually be done in the compressed domain and accomplished by manipulating the DCT and the MV components. Unlike transcoding process, extracting DCT and MV consumes only 10% of the resources that are required for full stream transcoding.

Our research to prevent such attacks is focused on preventing MV and DCT manipulation. The fundamental concept is based on random selection of MV and DCT coefficients and performing minor random changes of their values.

The prevention concept is essentially a self-immunization process by which an immune system becomes fortified against some types of malicious data (known as the immunogen). This process can be described as self-attacking with random parameters such that any attack will be impacted and destroyed by those random changes.

B. *Method*

The H.264 standard employs predictive MV coding technique using the median predictor of spatially neighboring MVs – Median Motion Vector (MMV). The motion compensated inter-frame prediction technique achieves high compression by reducing data, effectively exploiting temporal correlation, using accurate displacement (dx, dy) relative to MMV – Error estimation of MVs (See Figure 1).

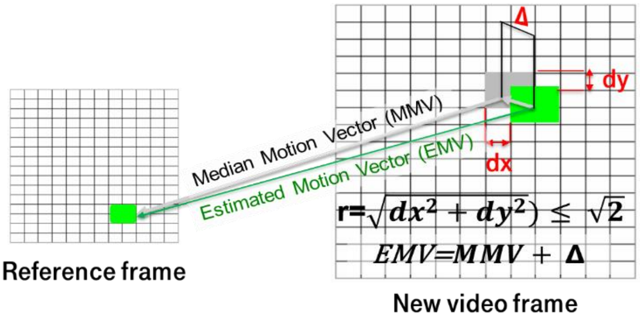
Δ
dy
Median Motion Vector (MMV)
Estimated Motion Vector (EMV)
dx
r=$\sqrt{dx^2+dy^2)} \leq \sqrt{2}$
EMV=MMV + Δ
Reference frame
New video frame

Figure 1: Median Motion Vector (MMV) and search criteria

The considered cyber-attack method is based on a search criteria that identifies best candidates for planting malicious data and replacing the original displacement (dx, dy) by the attacker’s malicious di-splacement (dx', dy').

The search criteria contains simple AND logic between two sub criteria:

1. Searching for long *MMV* that are longer than an arbitrary pre-define distance D_{MMV}.
   ($|MMV| \geq D_{MMV}$).
2. Searching for all MV displacements of size ($\sqrt{dx^2 + dy^2}$), that are smaller than a pre-defined threshold value (T_r).

(1) $r = \sqrt{dx^2 + dy^2}) \leq T_r$. *(e.g. r ≤* $\sqrt{2}$ *for dx, dy = 0, 1)*

The logic behind the search criteria is looking for very fast movements between frames, assuming that in such fast MB movements, the user will not be able to notice the noise, introduced to the frame by the attacker (inaccurate MB position compared to the original frame).

For example, in H.264 the Macro Block (MB) new position is expressed by the Error estimation of MV, compared to its neighbors median MV position. Table 1 provides available MB position coordinates relative to the center (0,0). The radius to all direction from 0,0 is smaller than $\sqrt{2}$. Let's assume that in a specific frame there is a MB with its MV a combination of its *MMV* and the displacement from its *MMV*, and the *MMV* size is longer then D_{MMV} (its neighbors are moving very fast) and its displacement is less than T_r (this MB is moving almost at the same speed and direction as its neighbors).

Table 1: Error estimation of MVs (displacement) smaller than $\sqrt{2}$

(-1,-1)	(0,-1)	(1,-1)
(-1,0)	(0,0)	(1,0)
(-1,1)	(0,1)	(1,1)

These nine (9) Error estimations of MV positions can be used as a dictionary for conveying malicious data. They can represent eight (8) different values (equivalent to 3 bits representation – 000 - 111) plus one additional value (the center – 0,0). Table 2 illustrates the 8 + 1 dictionary options corresponding to the above MVs' Error estimation.

Table 2: Dictionary options corresponding to the MV's Error estimation

7	0	1
6	X	2
5	4	3

Where X can be used as a special character, such as 'New Line', 'Space', etc.

For example, the encoder (or a proxy) that implements the algorithm will search in the frames for MVs' Error estimations which meet the criteria. Let's assume that the algorithm will find an Error estimation of MV (-1,1). Let's further assume that the attacker wants to convey the value '2' (010), this MV Error estimation will be replaced by (1,0) – See Table 1 and Table 2.

In the same example, the decoder will search for MVs' Error estimations that satisfy the same criteria. Once found, it will identify the (1,0) MV Error estimation and will convert it to the appropriate dictionary value (in this example - '2').

Our research and experiments demonstrate that an Error estimation of Motion Vector threshold criteria of $\sqrt{2}$ can support a covert channel transmission bandwidth of ~80kbs. This bandwidth was calculated according to formula (2).

Our proposed defense algorithm mitigates the attack by identifying and randomly changing the values of MVs' Error estimation that meet the criteria. By performing such a change, the covert channel is scrambled and the information therein is no longer valid.

$$(\,2)\; CCB = \frac{P*Fr}{FT} \sum_{i=1}^{N} Fi$$

Where

Fi=1 if MMV $\geq D_{MMV}$ *and r* $\leq T_r$ *, else Fi=0*

CCB – Covert channel bitrate
Fr - Frame Rate
FT - Total Number of frames in the movie
N - No. of MB in the all movie
P - No. of bits per element in the dictionary
MMV - Median of the neighboring MV
r - MV displacements size (r=$\sqrt{dx^2 + dy^2}$*)*

C. Research Structure and Lab Setup

The research program includes the following components:

3. Defense algorithm
4. Attack algorithm
5. Attack envelop (computers, smartphones, IoT)

As part of this research phase, we focus on attacks, initiated from within the LAN environment, thus performed from inside the organization. In order to evaluate covert channel available bandwidth and corresponding video quality degradation and in order to measure video delivery delays due to the attack, we created an attack envelop that uses ARP spoofing [8] for "hijacking" the user requested live channel video stream and replacing it by the infected one (see Figure 2 and Figure 3).

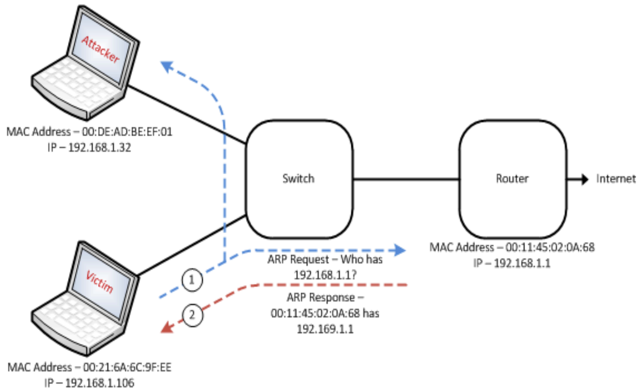
Attacker
MAC Address – 00:DE:AD:BE:EF:01
IP – 192.168.1.32
Switch
Router
Internet
MAC Address – 00:11:45:02:0A:68
IP – 192.168.1.1
ARP Request – Who has 192.168.1.1?
1
ARP Response – 00:11:45:02:0A:68 has 192.169.1.1
2
Victim
MAC Address – 00:21:6A:6C:9F:EE
IP – 192.168.1.106

Figure 2: Attack lab setup indicating stage A of the ARP spoofing used for hijacking the stream

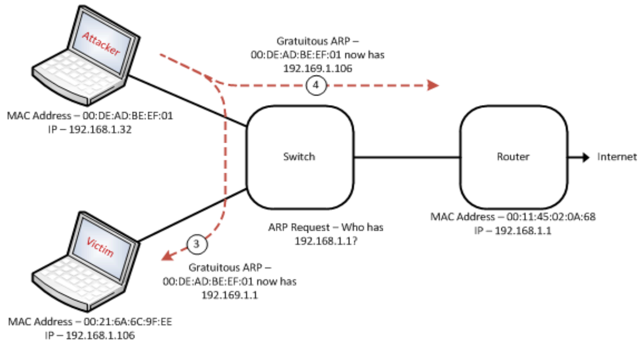

Figure 3: Attack lab setup indicating stage B of the ARP spoofing used for hijacking the stream

Our measurement results indicate that the delay introduced to the requested live video stream, due to stream hijacking, ranges between 10's to 100's msec, which is unnoticeable and may be attributed to regular network delays.

Using this method, some of the ARP updates will still arrive from the original real live streaming content server. Therefore, we can expect some temporary disruptions in covert channel transmission (when the user client switches back to the original server). The switch is transparent to the user and only means that covert channel will deliver its malicious content only part of the time, when it is the selected streaming server choice.

D. Implementation

We use this attack to route the victim's http request through the attacker who manipulates the data. The http video requests are filtered with IPTABLES (Linux networking tool) and changed to a different destination port (and/or address).

In our lab, we performed the fallowing Proof of Concept measurements, to understand the effect of a MITM (Man-in-The-Middle) on the innocent viewer.

1. Delay of the traffic routing through another computer between the Client and Video Server (Layer 3 re-routing only);
2. Delay of simple proxying with SOCAT (TCP listening and forwarding tool). (Layer 4 proxying only);
3. Delay of repacking the video (from one type of stream to another) via FFMPEG; and
4. Delay of full transcoding via FFMPEG.

Figure 4: Original Test Movie

E. Testing Setup

Test setup is described in Figure 5.

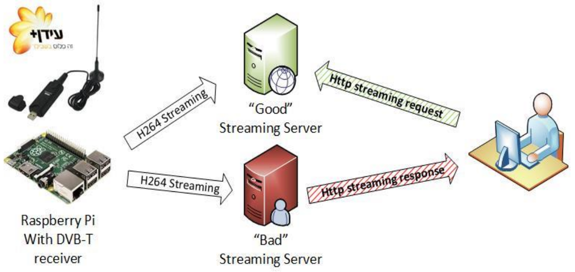

Figure 5: Laboratory Setup block diagram

The test setup included:

- A Raspberry Pi compute, producing H.264 video stream from a DVB-T source (Idan Plus);
- A simple HTTP Streaming server installed on Raspberry Pi (to accept and provide video requests);
- A malicious server to impersonate the server and achieve the MITM; and
- A VLC Client running on a PC ("Innocent victim").

All (Beside the Baseline) traffic has been re-routed through the malicious server with ARP spoofing MITM Attacked, as explained above.

The Measurements were taken in the context of live streaming user experience, hence we used two VLC players on the victim's computer, a clean video stream, one that was not re-routed through the attack server and a second video stream, which was routed through the MITM attack and manipulated according to the measured scenario.

The streaming video was timed to measure the delay between the two streams (the user is unaware of the exact time of the broadcast itself and is therefore oblivious to it).

IV. EXAMPLES

After investigating this algorithm on one specific video file (frames: 100, size: 360 x 640, frame rate: 25 frame/sec, overall bitrate: 2128 kbps, format: AVC/H.264, color space: YUV, Chrome subsampling: 4:2:0, Bit depth: 8 bits- See Figure 4), we found out that it is possible to create a covert channel of more than 3Kbps bandwidth inside that video. Changing the threshold value (and thus changing the number of selected MVs' Error estimation based on their size criteria) changes the covert channel bitrate. The value that determines the attack strength is the threshold level *Tr*. The lower the *Tr* value is, the less damage is caused to video quality. However, the covert channel bandwidth is also reduced as *Tr* decreases. Tests have shown that it is possible to increase *Tr* beyond $\sqrt{2}$ (which corresponds to 3 bits), such that more MVs' Error estimation are impacted, while introducing only a slight impact to video quality. Therefore, higher covert channels bandwidth can be achieved, while retaining reasonable video quality.

The *Tr* value can be modified for different purposes. Whenever security is more important than image quality, we would recommend using a high *Tr* value, while whenever security is less important than video quality, we would recommend using a low *Tr* value.

The picture of Figure 6 below illustrates varying number of MBs that meet the threshold criteria for different *Tr* values. The yellow squares indicate Macro Blocks with MVs' Error estimation that meet the criteria of being smaller than $\sqrt{2}$ (e.g., $|MV\ Error\ estimation\ | \leq \sqrt{2}$), while the red squares indicate MBs corresponding to a larger *Tr* value.

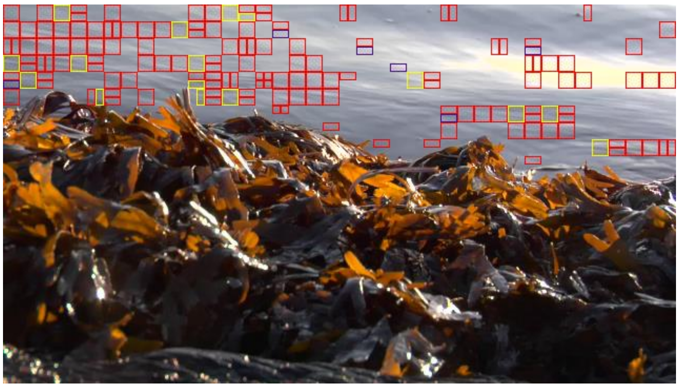

Figure 6: Picture illustrating different number of MBs that meet different threshold (Tr) criteria

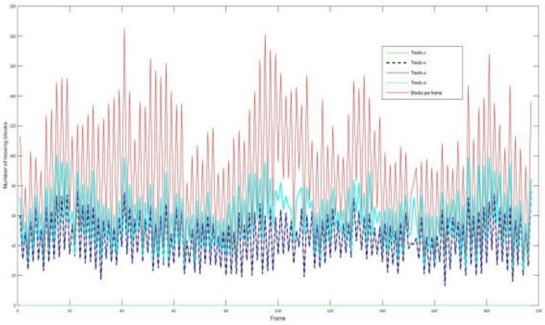

Figure 7: Graph indicating the smallest amount of vectors per frame for different threshold values

After inspecting the impact on video, we were able to create a cost-effectiveness graph of our entire defense algorithm, shown in Figure 7. In the graph we can see the number of MBs that potentially belong to the covert channel with respect to various values of *Tr*. The numbers also correspond to varying changes of MVs' Error estimation, minor in the case of small *Tr* values and higher as *Tr* increases. This approach allows users to perform risk analysis of their systems and set the acceptable quality degradation based on covert percentage.

Conclusions and future directions

In this paper, a unique technic was suggested for preventing cyber-attack using video streams. The new idea is based on manipulating motion vectors of compressed H.264 standard video streams. The paper is focused on prevention (not detection) of the malicious data from attacking the victim.

We have demonstrated that the attacker can achieve a covert channel of 80 kbps bandwidth. This technique can be accomplished by using proxies and in near real time, without the need to decode the video. Therefore, it will consume very low CPU processing time and can become very attractive to attackers. The protection technique is very simple. It uses the same technique as the attack to insert random noise at the client side. Client can choose their protection level. If they trust the video source they can decrease the *Tr* value and improve video quality. On the other hand, if they suspect the video source, yet insist to view the video content, they can increase the *Tr* value (while reducing the video quality) and watch it safely.

In future research, we plan to demonstrate an algorithm to prevent such attacks. The algorithm will work in a similar way to that of the attack, but unlike the attack, the prevention will pick random motion vectors and will change them randomly without significantly reducing the video quality.

Acknowledgment

This work was supported by the Israel National Cyber Bureau. The authors gratefully thank Michael Edry and Liran Odiz for implementing the attack algorithm.

References

[1] David Schneider, "The state of network security," *Network Security*, 2012 (2), pp.14-20.

[2] Andreas Neufeld, Andrew D. Ker Proc, "A study of embedding operations and locations for steganography in H.264 video,". SPIE. 8665, *Multimedia Watermarking, Security, and Forensics* (2013).

[3] T. Morkel, J. H. P. Eloff and M. S. Olivier, "An Overview of Image Steganography," Proc. *the Fifth Annual Information Security South Africa Conference* (ISSA2005) (2005).

[4] Amsden, N. D., Lei Chen, Xiaohui Yuan, "Transmitting hidden information using steganography via Facebook," *International Conference on Computing, Communication and Networking Technologies* (ICCCNT) (2014).

[5] Harsh K Verma, Abhishek Narain Singh, Raman Kumar, "Robustness of the Digital Image Watermarking Techniques against Brightness and Rotation Attack," *International Journal of Computer Science and Information Security*, IJCSIS, Vol. 5, No. 1 (2009).

[6] Lu Jianfeng, Yang Zhenhua, Yang Fan, Li Li, "A MPEG2 Video Watermarking Algorithm Based on DCT Domain," *Digital Media and Digital Content Management (DMDCM) (2011)*.

[7] Y. Amsalem, A. Puzanov, A. Bedinerman, M. Kutcher, O. Hadar, "DCT-based cyber defense techniques," Proc. SPIE 9599, *Applications of Digital Image Processing* XXXVIII, 95991F (September 22, 2015)

[8] Stefan Fouant, (Nov 2010) "Man in the Middle (MITM) Attacks Explained: ARP Poisoning," ShortestPathFirst (November 18, 2010)