Poster: LED there be DoS: Exploiting variable bitrate IP cameras for network DoS

Emmanuel Goldberg*, Oleg Brodt*, Aviad Elyashar*†, Rami Puzis*
*Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel
†Computer Science, Shamoon College of Engineering, Beer-Sheva, Israel

I. Introduction

Variable-bitrate (VBR) video streaming is widely used in IP surveillance cameras to balance video quality and network bandwidth conservation. However, VBR cameras can generate significant traffic spikes based on visual input dynamics. They convert light into electronic signals to capture images or video footage. Video recording is a sequence of 10-120 frames per second (FPS). Streaming these frames generates substantial traffic volumes. For example, an uncompressed 1080×1920 video at 60 FPS requires $\approx 2.986Gb/s$, making raw storage and transmission impractical. Video compression techniques reduce the data volume while maintaining a balance between visual quality and computational efficiency [2]–[4]. Common video compression standards (e.g., H.264/H.265), employed in IP cameras, assume that the captured scene remains primarily static over time. This assumption is a significant factor in reducing video transmission bitrate and bandwidth provisioning. A threat we explore here challenges this assumption. We demonstrate how the variable bitrate mechanism of video encoding can be exploited to generate traffic spikes and trigger a network denial of service (DoS) through changes in the camera's field of view (FoV).

Video encoding exploits the fact that our eyes are more sensitive to brightness (luminance) than color (chrominance), reducing color resolution via *chroma subsampling*, where color data are sampled at lower resolution than brightness. Another aspect is exploiting redundancies within and between frames. Spatial redundancy arises from similarities between neighboring pixels in a frame, while temporal redundancy stems from similarities between consecutive frames, especially in scenes with minimal motion. Instead of fully encoding each frame, only the differences between frames are encoded.

Regular motion events in an office environment have a negligible impact on the stream size, as modern video compression algorithms utilize motion estimation and compensation techniques. Motion estimation identifies blocks of pixels in one frame corresponding to blocks in a reference frame. The displacement of these blocks is represented by *motion vectors*, which describe the direction and magnitude of movement. Motion compensation uses these vectors to predict the content of the current frame based on reference frames. The data volume is significantly reduced by encoding only the differences between actual and predicted frames. For instance, a person walking in a corridor impacts the generated traffic similarly to a person standing in place. Therefore, unexpected or unusual changes in the field of view are required to generate excess traffic.

Here, we show how an adversary can compromise the effectiveness of video compression by introducing rapid visual changes using a flickering laser LED. Such a laser, pointed at a variable-bitrate IP camera, can force it to generate 5- 6x more bandwidth. Preliminary experiments with IP cameras in both wired and wireless network setups indicate that the laser attack may cause significant packet loss in poorly planned networks.

II. Threat Model

In this attack, a threat actor would like to disturb communication in the victim network, potentially causing a DoS. We assume that the victim's computer network is air-gapped, and there is no easy way to obtain a digital footprint within the network. The victim employs high-resolution video surveillance for physical security and to track movements within the facility. We assume the victim employs the same physical communication infrastructure (routers, switches, cables) for video streaming traffic and other IT/OT operations. Of course, there is complete logical segregation between the security surveillance network and the facility operations. No communication is possible between the segregated networks, but they compete over the same shared network resources. We assume 90%-95% peak utilization of the local network bandwidth and that the adversary may have a good estimate of the timing of such peaks. Finally, we assume that the adversary is in sight of one or more IP surveillance cameras and has a strong enough laser to reach the cameras from the required distance.

III. High-level Description of the Dazzling DoS Attack

The adversary may execute the attack by pointing a laser flicker beam toward one or more IP cameras within the victim's premises. The affected cameras identify changes in lighting caused by the laser as the scene changes, generating additional IP traffic. The excessive traffic traverses the network to reach the camera viewing station. Since the traffic passes through shared physical resources (devices or links), their overload can lead to congestion and packet drops.

To increase an IP camera's bitrate to the extent required to ensure the attack's success, we should generate a signal that modern video encodings cannot efficiently compress. Algorithms for this purpose perform many optimizations to

reduce the video's bitrate to the greatest extent possible without harming quality. However, rapid unnatural changes in the scene reduce the compression quality to a great extent. To generate such rapid changes, an attacker located at a camera's direct line of sight can dazzle it using a flickering laser beam. As detailed in [1], dazzling an IP camera with a laser has many physical effects, including light diffraction, reflections, and scattering of the laser beam. This attack requires no digital foothold on the device or network, exploiting the inherent link between physical phenomena and digital behavior in cyber-physical systems.

IV. Experimental Design

To assemble the attack device, we used a simple LED laser diode. The laser is controlled by an Arduino. To fix the laser position and direction, we mounted it on a camera tripod commonly used by photographers. Although the Arduino can turn the laser on and off, the frequency of doing so is not high enough. A fan in front of the laser can generate very high frequencies by blocking the beam with its wings. Using a fan to create the flicker has an additional (not-so) surprising effect that increases the video streaming bitrate. A small servomotor powering a fan mounted on the tripod in front of the laser emitter generates small vibrations that significantly affect the light reflections and scattering. The small vibrations of the motor create slight movements without causing the laser to lose focus on its target. These vibrations introduce additional chaotic behavior to the captured frames, further increasing the frame-to-frame differences and hence the bitrate. While our implementation uses a fan to generate rapid fluctuations, we note that commercially available high-frequency modulation circuits (e.g., PWM-controlled laser drivers) could achieve similar results in a more controlled and compact form. Our goal was to demonstrate feasibility using low-cost, easily accessible components.

To evaluate the effectiveness of the attack on the network that the affected camera is connected to, we constructed both wired and wireless network testbeds: the camera was cable-connected to the network, while the rest of the network could be wired or wireless. During evaluations, we generated different constant network loads that simulate baseline activity on the network and different network load profiles for network disruption testing.

We also created a list of disruption metrics. Since different networks may have different traffic types as their baseline, we wanted to ensure that the metrics created apply to various common network scenarios.

Finally, the attack was also tested in settings where the laser was at various angles with the camera.

V. Results

1) Impact on Camera-Generated Traffic: Overall, using the laser flicker, we achieved a VBR bitrate increase of a single camera of around $6\times$, up from 2.25 Mb/s to 13.5 Mb/s. When changing the horizontal angle, we found that the increase in bitrate diminishes as the incidence angle increases. However, when changing the vertical angle, the bitrate remains fairly constant before reaching an abrupt cutoff at ˜45°.

2) Impact in a Wired Setting: The attack caused significant network disruptions. During the attack, RTTs increased dramatically, reaching up to three seconds, along with a packet drop rate of 10–20% under optimal conditions and up to 80% in more demanding settings. These results were consistent across TCP and ICMP traffic.

Similarly, the file transfer and TCP stress test metric showed up to a 90% reduction in effective throughput, largely due to congestion control mechanisms, while the UDP stress test metric showed a 5–10% decrease. This is proportional to the video stream's share of overall traffic and reflects the protocol's lack of congestion management.

3) Impact in a Wireless Setting: The disruptions were similar to those in the wired setup, with additional vulnerabilities due to the shared nature of transmission queues. In the round-trip ICMP and one-way TCP data metrics, the results were broadly similar to the wired setting, with higher RTTs across the board and even some drops in the most aggressive baseline scenarios. A notable difference is that in the wireless setting, ICMP traffic struggled more than TCP traffic, as it had to traverse the overloaded transmission queue twice.

Due to the already stressed transmission queue, sending large volumes of traffic became tediously slow before initiating the attack and nearly impossible during it, and so only the TCP and UDP stress test metrics were benchmarked for the lightest base-load setting, in which we see a 90% reduction in TCP traffic and a 75% reduction in UDP traffic. Compared to the wired setup, the notable reduction in UDP traffic is primarily caused by the attack's ability to increase the load on the shared wireless channel, meaning less UDP traffic may be transmitted to the router.

4) Defensive Considerations: A potential countermeasure to this attack is a low-frequency filter applied to the camera input, effectively ignoring high-frequency visual fluctuations. While theoretically appealing, such filtering could degrade responsiveness to legitimate fast-changing scenes (e.g., flashing alarms, strobe lights) and may not be effective against patterned or modulated attacks designed to evade static thresholds. A more robust defense likely requires adaptive filtering or firmware-level traffic shaping.

References

[1] C. Booth and M. Richardson. Visible and near-infrared laser dazzling of ccd and cmos cameras. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 10797, page 107970S. SPIE, 2018.

[2] Keith Jack. *Video Demystified: A Handbook for the Digital Engineer*. Elsevier, Burlington, MA, 2011.

[3] Djordje Mitrovic. Video compression.

[4] Ben Nassi, Raz Ben-Netanel, Adi Shamir, and Yuval Elovici. Game of drones - detecting streamed poi from encrypted fpv channel, 2018.

LED there be DoS:
Exploiting variable bitrate IP cameras for network DoS

Emmanuel Goldberg, Oleg Brodt, Aviad Elyashar, Rami Puzis
Ben-Gurion University of the Negev, Shamoon College of Engineering

1. Highlights

- Improper network design and deployment of IP cameras can be exploited by attackers to cause DoS.
- By challenging the assumption of scene stationarity, an attacker can cause variable bitrate IP camera generate excessive traffic without any digital footprint within the victim organization.
- A simple setup with a flickering laser diode generates rapid changes in the scene which reduce the compression ratio of common variable bitrate video codecs.

2. Research Questions

- What is the impact of a flickering laser diode on the IP camera bit rate and overall communication network performance?
- How does the effect of excess camera traffic on network performance differs in wired and wireless setups?

3. Background

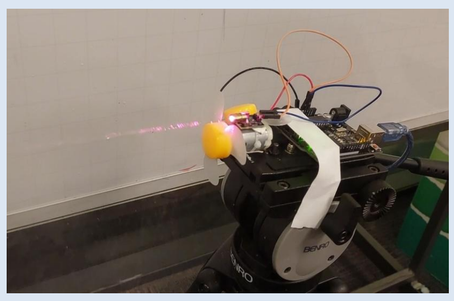
- Video streaming cameras are an integral part of modern life. Scattered across cities, office buildings, transportation hubs, and critical facilities, they monitor the environment for hazards, suspicious activities, quality control, etc.
- Transmitting full high-resolution frames at high frequencies (e.g., 30 or 60 FPS) for a static scene is wasteful. Therefore, modern surveillance cameras are variable-bitrate devices. In addition to compressing individual frames, video codecs in IP cameras reduce bandwidth usage by transmitting mainly the changes in the captured scene.
- VBR cameras operate under the assumption that the scene they capture remains mostly static over time. They can therefore generate sharp traffic spikes depending on the dynamics of the visual input.

4. Effect of Laser Flickering on Video Compression

Video Compression Optimization	Laser Flickering
Color information is compressed, while dark-bright contrasts are not.	Laser diffraction yields very uniform colors, as well as many dark-bright contrasts, which are further exacerbated by the laser's flickering.
Recent past (and future) frames are used to encode the current one. Specifically, the movement of an object from one location to another is encoded as a motion vector rather than as a retransmission of the object.	Frames vary wildly from one to the other, and these variations cannot be described or even approximated as linear movement.

5. Testbed Setup

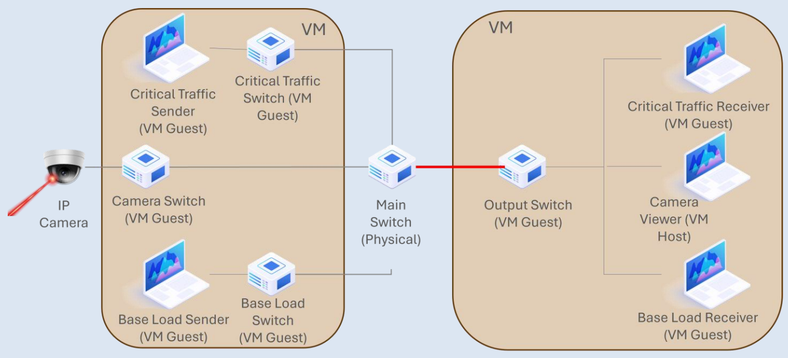
Laser Mount



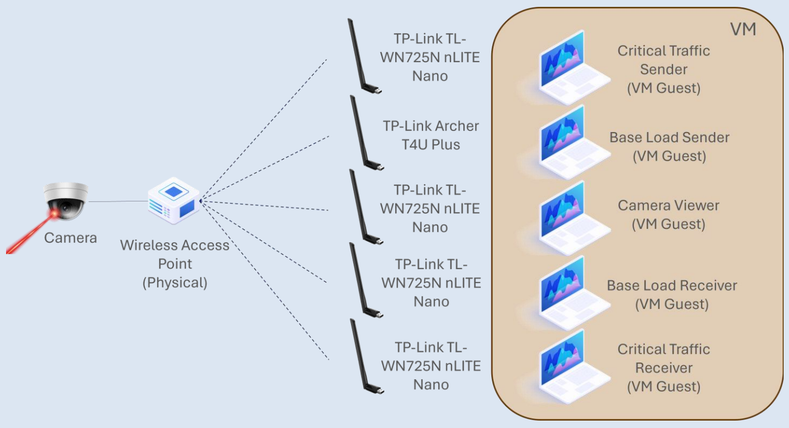
Dazzling the camera

Camera's FoV

Wired Setting



Wireless Setting

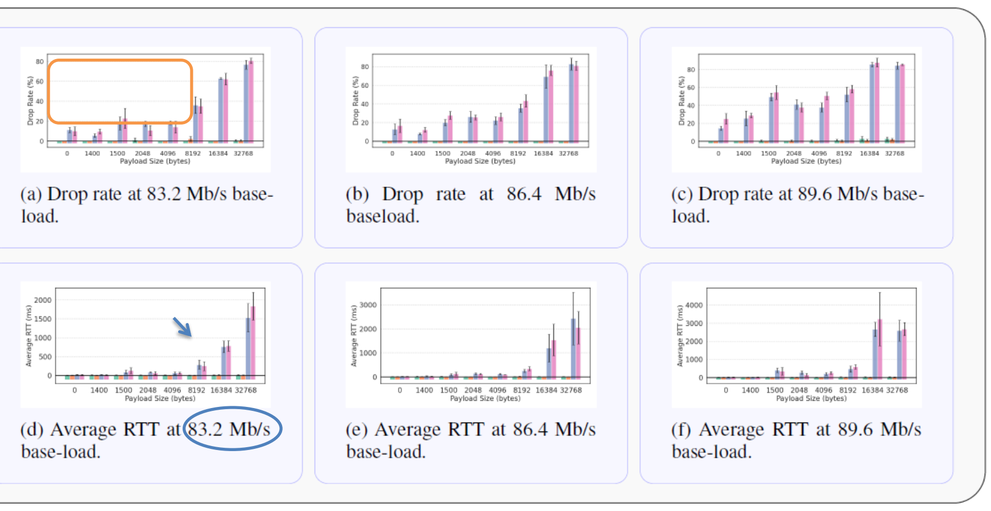


6. Evaluation Metrics

Name of Metric	Tool Used	Description	Purpose
Roundtrip ICMP	hping3 (ICMP)	Transmit ICMP echo requests and receive echo replies back	Test dropped pings during data exchange
One-way TCP data	hping3 (TCP)	Transmit TCP data and receive an ACK back	Test dropped pings for one-way data transfer
File transfer	rsync (SSH)	Transmit a 10 MB file over SSH	Real-world network bandwidth test
TCP stress test	iperf3 (TCP)	Attempt to transmit 100 Mb/s TCP traffic for 60 seconds	Measure bandwidth with congestion control
UDP stress test	iperf3 (UDP)	Attempt to transmit 100 Mb/s UDP traffic for 60 seconds	Measure bandwidth without congestion control

7. Results

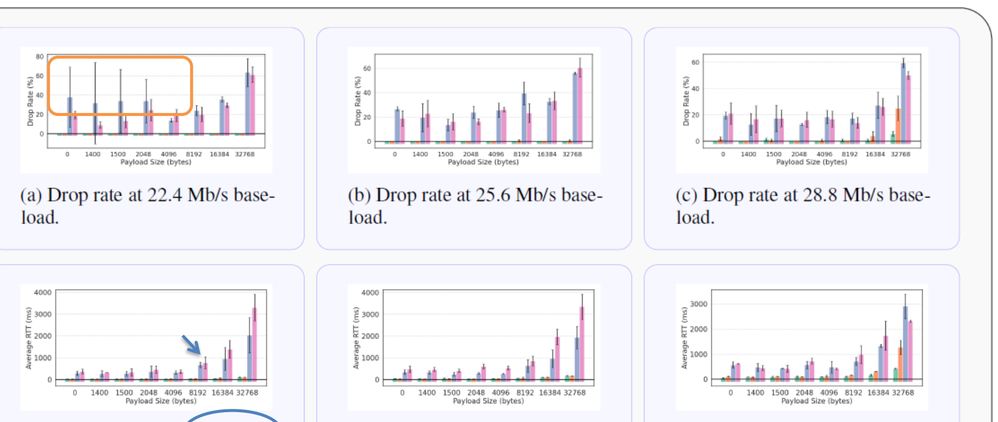
Packet loss and latency increase in wired setup



(a) Drop rate at 83.2 Mb/s base-load. (b) Drop rate at 86.4 Mb/s baseload. (c) Drop rate at 89.6 Mb/s base-load.

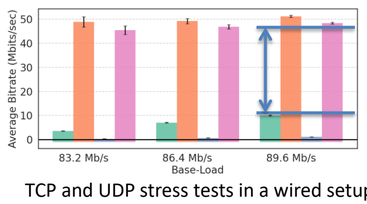
(d) Average RTT at 83.2 Mb/s base-load. (e) Average RTT at 86.4 Mb/s base-load. (f) Average RTT at 89.6 Mb/s base-load.

Packet loss and latency increase in wireless setup

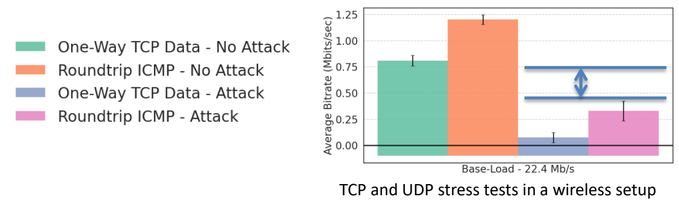


(a) Drop rate at 22.4 Mb/s base-load. (b) Drop rate at 25.6 Mb/s base-load. (c) Drop rate at 28.8 Mb/s base-load.

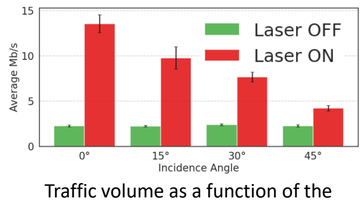
(d) Average RTT at 22.4 Mb/s base-load. (e) Average RTT at 25.6 Mb/s base-load. (f) Average RTT at 28.8 Mb/s base-load.



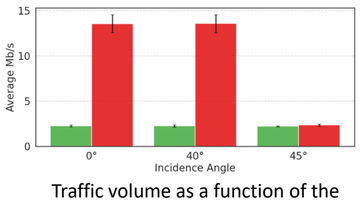
TCP and UDP stress tests in a wired setup



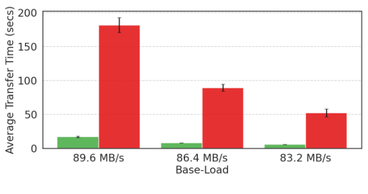
TCP and UDP stress tests in a wireless setup



Traffic volume as a function of the horizontal incidence angle



Traffic volume as a function of the vertical incidence angle



File transfer in a wired setup