[2135] A Bad IDEa: Weaponizing uncontrolled online-IDEs in availability attacks

PRELIMINARY DECISION: accept

Summary of Reviews

- Review 1: 1 (4)
- Review 2: 0 (4)
- Review 3: 0 (4)
- Review 4: 3 (3)
- Review 5: 0 (5)

Reviews

Review 1

TOTAL SCORE: 1

Overall evaluation: 1 (weak accept)

Reviewer's confidence: 4 (high)

This paper presents a study of online IDE platforms, specifically oriented to assess if these have uncontrolled execution environments, and how they can be abused for malicious purposes. The research involved active exploitation/abuse of those online IDEs that consented to the research. To prove the feasibility of the abuse and estimate the possible impact, authors developed their own botnet and performed short attacks.

The research is very interesting, novel, and the impact seems clear. It is however not fully connected to the reality of actual cybercrime operations. There's no evidence of such attacks actually taking place in real life, giving the conclusions of the research statistical and theoretical value and not an actual measurement of cybercrime operations which is the primary focus of WACCO.

Specific comments:

- The paper structure could be improved as methodology with experiments are somehow mixed together in Section 4.

- While there's a clear description on how Google was leveraged to find new IDEs, it is not clear how Shodan and Censys were used as they are generally used for other purposes.

- The requests for consents is not clear. A small subsection on this aspect would be recommended. The 2k IDEs were contacted? How many did not have contact details? How many answered? How many did not answered? How much time it was given for them to answer?

- While fully valid, it not fully clear why the HTTP flooding was chosen specifically among other type of attacks that could be carried out.

- The bot attack code was obtained from GitHub and modified to match the research requirements. There is no citation or references to the original code.

- When discussing the multi-vector attacks in Section 6.1.3, is not clear why some IDEs are not vulnerable and others are. Other attacks are explained or easily understood, however this is not trivial to the reader.

- Section 6.1.4 discusses Text-encoding attacks. How are these attacks important on this scenario? Most online IDEs do not seem to have checks in place that could seem to detect malicious code, so how text-encoding would help?

- Overall the connection and references to malware injections, botnets, and cryptomining through the work are a bit weak. To make a stronger connection to the presented research, a section discussing two or three particular cyber threats and how specifically they could leverage these Online IDEs would be better than loosely mentioning malware across the text.

Review 2

TOTAL SCORE: 0

Overall evaluation: 0 (borderline paper)

Reviewer's confidence: 4 (high)

This paper analyses the usage of online IDEs as methods to perform DoS attacks to other systems. The paper has several contributions ranging from a definition of what a vulnerable IDE is, to the construction of a system that can be used to launch a coordinated DoS attack via HTTP requests from online IDEs. The authors perform an experimental evaluation and then perform some analysis on the number of requests that could be sent via this method.

The paper proposes a very interesting and promising idea but I think suffers from a series of issues.

Strengths:

- The idea is novel and quite original

- Authors did responsible disclose the issues they found and only conducted experiments with permissions

- Authors mention the limitations of their work and propose countermeasures

- Authors build a framework that could be used by an attacker to build a more sophisticated attack instead of using a single online IDE showing how this can be done at scale.

Weaknesses:

- The definitions of Uncontrolled IDE environments are quite vague and in some cases, they even don't have anything to do with the actual DoS attacks. For instance, environments may allow unrestricted file operations but within a limited VM or infrastructure. This is not necessarily wrong and it doesn't affect the attack that is presented later. Authors should focus on the aspects that affect the attacks they build.

- The checks performed by the 'crawler' are not really clear or well defined. A list of specific checks should be provided in the appendix to actually verify that those checks are in line with the properties that are being searched for. For instance, what is the list of restricted packages or modules that should not be imported? Also, why use docker to check for sandboxed environments? Is that the only way of checking?

- I am also not sure why the authors call their experimental setting a botnet architecture. The experimental settings show a set of IDEs being controlled by a Selenium bot that is in charge of organising the executions, etc. but this is far from what a regular botnet architecture would look like (each bot installed on a different machine, etc.). It is not clear to me how this translates to a real botnet infrastructure. This is, the authors may be able to create a few DoS bots but it is important to note the limitations of this botnet in terms of updates, etc.

- Figure 5 is confusing as the categories are not mutually exclusive (shown by the presence of all).

- I understand the difficulties of performing these tests without compromising the security of the systems being evaluated and I praise the authors for seeking permission from providers. I found it underwhelming that the average number of requests per IDE session of 5 minutes was 103. This is a request every 3 seconds. How does this relate to Figure 6 where there are thousands of requests per second in the graph? If the latter a simulated then the authors should clearly state that their experiments are very limited and are just proof of something being possible (more on this later).

- The comparison between language IDEs seems flawed to me as well. Aren't all IDEs running the same algorithm to send packets? If that is the case, they should all be sending the same number of packets right? If this is true then the difference in the numbers may be because of network conditions but have nothing to do with the security. It only talks about the conditions of the networks when these tests were executed.

- In fact, this is something that is discussed later on the limitations sections. This section covers a wide range of issues, including some of the ones that I mentioned earlier. However, some of those issues should be discussed in the context of the results and not in such a summarised way in a small section at the end. The countermeasures proposed are quite straightforward and are also not discussed in the context of the research. For instance, the authors mention that they rate-limited their request because of ethical concerns but also say this as a countermeasure. If this is the case, how they can be sure that they didn't reach the rate limit of the analysed sites?

I think the authors should clarify some of these issues in their final version of the paper and be clear about the limitations of the experiments they perform.

Review 3

TOTAL SCORE: 0

Overall evaluation: 0 (borderline paper)

Reviewer's confidence: 4 (high)

The authors report on an interesting way of abusing publicly available services (online IDEs) to launch DDoS attacks and perform cryptomining. Through an exploration via a search engine a large number of online IDEs (719 of which had uncontrolled execution) were identified. For 18 online IDEs a more extensive evaluation on their susceptiveness for attacks was performed.

Overall the paper presents an interesting and detailed study on a highly relevant topic with real-world impact. Although the analysis is mostly sound and I appreciated the ethically-motivated approach of performing DDoS-attack estimates, it is unclear whether this method would be practical in a real-world DDoS attack. More precisely, one of the reasons that DDoS attacks launched from large botnets are effective is that these originate from a wide variety of IP addresses, making it difficult to block these. It would be interesting to explore from how many different source IPs the network requests were sent, and whether creating new instances resulted in new IPs.

It is not clear to me how the bandwidth (expressed in Mb/s) can grow over time, as shown in Figure 9. Furthermore, the reported 820Mb/s does not match with what is shown in the graph. For the comparison between UDP and HTTP requests, was the same length of payloads used? How often was a new connection created to send data?

The authors mention that running the attacks has zero costs. This seems incorrect as an attacker would still need to use a server with an instrumented browser to run the code on the IDEs.

Typo: 7.3.2 - "the the"

Review 4

TOTAL SCORE: 3

Overall evaluation: 3 (strong accept)

Reviewer's confidence: 3 (medium)

This paper looks at a novel aspect of the botnet/cybercrime infrastructure ecosystem - online IDEs, which are often left unsecured and can become part of the attack infrastructure. The paper has a decent background section, which covers the technical basics and former work well. It'd be nice to have a little more here on the criminological background if space allows - even a couple more references to the cybercrime ecosystem/exploit economy and crime-as-a-service paradigm would provide useful context. The background section generally sets up the rest of the paper, and why it is important, very well. Section 3 provides a useful bridge, and I particularly liked your elucidation of the attack/vulnerability types here.

In section 4, I really like your survey-style methodology via Google Dorking, but you could reflect more here on the bias this brings in - presumably this misses IDEs in, e.g. Chinese, Russian or Arabic? Maybe I'm wrong about this - but if I am, then a comment in the paper as to why would be useful. Equally, it'd be good to know - do you think botnet herders in the wild looking to exploit this would use a similar method to find exploitable IDEs? If so, is this automatable, and hence a point of potential mitigation? I really like 4.2 - this takes you beyond an abstract survey of exploitable IDE features, and into the realms of real testing, nice one. I'm glad to see you got consent as well!

The testing appears rigorous and ethically carried out - great. Your attack enumeration in 6 is good too, and it was nice to see consideration of the newest, up-to-the-minute attacks (like Boucher et al). It'd be interesting to know, regarding the comparison with amplification methods, whether these IDE botnets might also be honeypot-able (if they ever get exploited en masse through automated scanning) in the same way that amplification-reflection lists are - so setting up a small fake vulnerable IDE for measurement purposes?

One thing I might suggest adding is a couple of thoughts on what it might be like to run a botnet like this in practice. What would the administration side be like? Is this going to be easier, harder, more or less labour intensive than an existing botnet? Is it going to require different kinds of crime scripts, skills, approaches, etc.? And what about mitigation beyond the IDE side itself - are there useful new vectors for takedowns or mitigations outside the IDE itself you can think of which would work here? None of this necessary, but might be useful to think about?

Some drafting errors that need to be picked up by another copy edit pass (including the first sentence, which is missing a verb). Otherwise, a great paper and a clear candidate for inclusion in WACCO.

Meta Review

TOTAL SCORE: 0

Overall evaluation: 0 (borderline paper)

Reviewer's confidence: 5 (expert)

All reviewers found the paper to present an interesting idea and results, and praised the adopted ethical approach. However all reviewers remark that some of the claims in the paper are not realistic, particularly in the limitations of the work in real-world settings.

The authors are asked to address the comments in the reviews before submission, with particular attention to the discussion of limitations of their work in the real-world, and to removing over-claims in the paper as underlined in particular by R2 and R3.