NETACEA / REPORT

NETACEA | BLADE

Business Logic Attacks and the BLADE Framework:

A Framework to Facilitate Detecting, Interpreting and Mitigating against Business Logic Attacks Launched over the Internet

AUTHORS: MATTHEW GRACEY-MCMINN, LIAM JONES, & ANTONY BARNETT

ABSTRACT

In this paper we highlight the growing threat of business logic attacks as a methodology by which adversarial actors are exploiting web applications and APIs for gain. For most businesses this is an area of increasing concern, but defenders do not yet have a coherent response by which to defend against these attacks, with no standardised use of language across the cybersecurity industry, nor a consistent, repeatable methodology to assist in interpreting and understanding these attacks. Consequently, defences are largely reactionary, and defenders have a limited understanding of impact and risk as an attack progresses. We believe that the Business Logic Attack Definition Framework (BLADE Framework) plugs this gap, by providing a comprehensive framework that details the tactics, techniques, and sub-techniques employed by any variation of a business logic attack during each of the stages it passes through. This can then be employed for: better threat modelling, improved risk assessment, strengthened detection and mitigation capabilities, increased ROI on cybersecurity investment, a better-informed incident response process, and improved reporting. By making this an open-source framework we hope to drive improvement in defences for all businesses impacted by such attacks and solicit feedback in improving it and ensuring it continues to be a relevant and useful tool for defensive teams.

BUSINESS LOGIC ATTACKS AND THE CYBER THREAT LANDSCAPE

The cyber threat landscape is ever evolving. Such evolution is driven largely by adversaries whose adoption of new technology, new tactics, new techniques, and new procedures requires defenders to adapt our own understanding and defensive procedures at a rapid pace. Historically, much defensive attention has been directed towards preventing what we, in this paper, term 'technical attacks'; that is, those attacks which seek to exploit code vulnerabilities or perform attacks against systems and networks. However, there is now a significant need for defenders to consider 'business logic' attacks. That is, those attacks which do not seek to attack the underlying systems or code, but instead use a web application or API in its legitimate, intended fashion to undertake malicious activity. These attacks are almost always at least partially automated and conducted by 'bots', tools that perform automated actions. The simple nature of business logic attacks lends them nicely to automated tools, which can conduct such actions repeatedly across multiple targets and at speeds far in excess of what a human can achieve. Contrary to traditional, technical attacks, business logic attacks can easily be launched by very low-skill adversaries, at very low cost. They nonetheless can have a major impact on businesses, causing significant losses and loss of competitive advantage.

Perhaps the most well-known example of a business logic attack would be an Account Takeover (ATO) attack, in which an adversary uses stolen login details to access the account of an unfortunate victim (bots facilitate this process by such actions as credential stuffing into login forms). The attacker has provided the correct login credentials and so is provided with access.

Hence, the webapp in question is functioning as intended by permitting access to someone who provides the correct login details. However, ultimately, this attack could have far-reaching impact for both the victim and the targeted organisation, with the victim suffering:

- The loss of payment details
- The loss of personally identifiable information (and potentially access to the service associated with the account).

Meanwhile the company may incur:

- Costs through data protection fines or payment card chargebacks
- Loss of brand reputation (often the targeted organisations are blamed for these attacks in the public sphere)
- Loss of customer confidence
- Secondary costs incurred indirectly in time and resources spent trying to 'repatriate' breached accounts to their correct owners.

A similar attack can be observed in the case of 'scalping', wherein an adversary identifies a high-demand product which will likely have significant resale value. They then use bots to identify the exact moment that this product becomes available on an online store and complete the purchase of a significant proportion of the overall stock far faster than any human could hope to perform the same action. More advanced bots will then automatically relist the product for sale at a marked-up price elsewhere, or even hold the product reserved in a cart until the resale can be completed and the cart handed over to the purchaser at a cost above that of the product. While not as directly impactful as either the above ATO attack example or more technical attacks that may bring down the webapp entirely, scalping attacks can have the following effects:

- Loss of custom (customers faced with being unable to purchase desired items may take their business elsewhere on a permanent basis)
- Loss of brand reputation (many of the scalper groups boast of successful 'scalpings', and organisations unable to fend them off are often regarded poorly)
- Increased cost of serving traffic (bots perform very aggressive scans when looking for the exact moment a product becomes available, often generating more than 80 times the amount of traffic a human would generate if simply refreshing a single page over and over)
- Loss of sales (where bots simply hold a product in reservation until a new buyer can be found there is an increased risk of no sale ever being completed, with legitimate customers being unable to purchase from the store while the scalper locks down a product they will not purchase)
- Ever-escalating scalping (as profits increase so too does competition between bot users who thus become increasingly aggressive in targeting sites, this is exacerbated by the growing number of people choosing to employ scalping tools as the only means to acquire the products they desire).

A 2021 survey of 440 businesses in the UK and USA conducted by Netacea[1] found that there is a growing concern within businesses as to the financial impact of these attacks. As noted above, these business logic attacks are often conducted via automated tools, which are known as bots. Many of these bots are categorised according to the nature of the attack they perform and their objective. The survey focussed on asking about specific types of bots and their impact on businesses. Some of the key findings were as follows:

In 2020, 65% of the businesses had detected bot attacks against their website, 46% had observed bot attacks against their mobile application, and 23% had identified bot attacks against their API.

- 85% of businesses reported an increase in the number of bot attacks against them in 2020 compared to 2019.
- Of the 440 businesses only 2 did not have a dedicated budget for bot management. 10 businesses reported it was more than 20% of their overall security budget.
- Of the businesses impacted by scraper bots[2]:
  - 15% had lost more than 5% of their total online revenue due to these attacks.
- 48% of businesses lost 3% or 4% of their online revenue due to these attacks.
- Of the businesses impacted by scalper bots[3]:
  - 21% of businesses reported losses of 5% or more of total online revenue because of these attacks.
  - 50% of businesses reported losses of at least 4% of total online revenue because of these attacks.
  - No business reported losses of less than 2% of total online revenue because of these attacks.
- Of the businesses impacted by account takeover bots[4]:
  - 21% reported losses of more than 5% of total online revenue due to these attacks.
  - 82% reported losses of at least 3% of total online revenue because of these attacks.

Footnotes:

1.The surveyed businesses came from Travel, Online Gaming, eCommerce, Financial Services, and Telecommunications industries. Surveyed businesses varied in size from $350 million to over $7 billion in size. All the answers pertain to the 2020 calendar year.

2. Scraper bots are those which are used to repeatedly scan a webapp or API and collect information from the contents of the page.

3. Scalper bots automate the purchase of low-supply, high-demand items such as concert tickets, video game consoles, limited edition clothing items, etc., which can then be resold at a profit by the bot user at a marked-up price.

4. Account takeover bots attempt to attain access to accounts of other users of a webapp or API by either using stolen or guessed credential pairings. Once they attain access, they are able to act as though they are the legitimate owner of the account and assume control of it.

Business logic attacks are distinct from the more widely observed and analysed 'technical attacks' but are similarly, if not more, pervasive. They target web-facing services that organisations offer to their customers, most often web applications, and APIs. They demand very little technical skills of those conducting them and are often very cheap to launch. Quite often they are also legal or, due to the difficult nature of attribution and the commonplace nature of such attacks, highly unlikely to lead to any sort of action by law enforcement against the perpetrators. This has made them a tempting attack methodology, particularly during the COVID-19 pandemic and the global shift towards an online-first social model, which drove many more services and businesses online than ever before. The increase in online business activity increased the number of potential targets and victims, and the profits to be made from such attacks. Consequently, we have observed a significant increase in the quantity of such attacks, as well as the sophistication of adversaries as they compete with one another to profit most from these attacks.

These 'business logic' attacks are a growing concern for many organisations, and yet, as an area of cybersecurity, the understanding and modelling of the threat landscape lacks the maturity of that surrounding the more traditional 'technical' attacks. There are theoretical models for the more traditional, technical attacks that can be used by defenders to understand what stage the attack is in, what techniques an attacker is likely to employ, and what may be attempted next. Such models include the Lockheed Martin Cyber Kill Chain and the Mitre ATT&CK Framework. These are widely employed across the cybersecurity industry, but efforts to apply them to business logic attacks routinely fail. This is because, as noted before, business logic attacks are not seeking to exploit systems maliciously or use them illegitimately and are instead employing them in their intended manner.

Efforts to respond to these business logic attacks have been hampered by a lack of understanding of how the attacks progress. Some efforts have been made in this area, most notably with the OWASP Automated Threats (OAT) project.[5] This provides high-level details of the sorts of automated attacks that often exploit business logic but does not give a granular assessment of the stages each attack type goes through and thus does not prepare and inform defenders in the same manner as the Mitre ATT&CK Framework does for other types of attacks. In this paper we posit that a new framework is necessary, one similar in style to the Mitre ATT&CK Framework but specifically detailing business logic attacks. Like Mitre this should detail the common tactics, techniques, and sub-techniques employed by attackers. It should also allow for all variations of business logic attacks to be mapped to it, facilitating an understanding of the stages different types of attacks go through, and where best to break the attacks' 'chain' to maximise defensive capabilities while minimising costs and risk.

Footnotes:
5. This project is the work of Tin Zaw and Colin Watson.

THE VALUE OF A BUSINESS LOGIC ATTACK FRAMEWORK

Through the examination of datasets linked to historical attacks, conversations with experts, and reference to adversaries' discussions and operations (which was achieved by infiltrating their online forums and group chats), we have constructed and tested a framework that meets these requirements. Taking inspiration from the Mitre ATT&CK Framework, it outlines the tactics, techniques, and sub-techniques commonly employed by adversaries when attacking via a business logic attack vector. The stages that different types of business logic attack go through can be charted across the framework, to build a 'kill chain' for each type of business logic attack.[6] We have found the BLADE Framework supports an intelligence-driven response to business logic attacks. In testing it has provided the following benefits:

Codification of business logic attacks:

Business logic attacks are experienced by many organisations, but there is no standardised language. Instead, each organisation has its own language and terminology. This complicates collaboration and the sharing of threat information. By codifying these attacks and standardising the language used we can facilitate inter-organisational collaboration.

Detection:

Rather than examining data sets looking for anomalies or known-bad technical indicators (such as malicious IPs), we can now look for tell-tale signs of specific tactics, techniques or sub-techniques. This allows responders to detect based on tactical-level information rather than the easily changed technical indicators of an attack.

Mitigation:

By establishing how each technique works we can move from a model of simply trying to block bad actions, to better understanding and interpreting what is being seen and taking steps to minimise risk by interrupting an attack at the most appropriate moment. This minimises both risk and the effort (and thus cost) of mitigating attacks. It also allows for more precise mitigation efforts; for example, blocking bad IPs may bar legitimate users from accessing a service, but by mitigating against specific techniques we are able to deny attackers without disrupting legitimate users of a service.

Better Understanding of Risk:

By understanding the stages an attack goes through, we can better understand the likely next steps of an attacker and the impact of each technique as it is employed.

Threat Modelling:

In building web-facing applications and APIs, we can refer to the BLADE Framework to better understand what sort of business logic attacks may be employed against them and so implement security measures against such attacks during the development cycle.

Footnotes:
6. These kill chains can be found at the BLADE Framework's website: www.bladeframework.org.

Lessons Learned:

After an attack is detected (whether that is during the attack or after it is complete), the framework provides a more granular break down of likely steps taken by the attacker during each stage of the attack, including undetected stages. This allows responders to go back through data sets to identify the earlier indicators and use these to improve future detection and mitigation efforts.

Reporting:

A granular understanding of attacker methodologies, kill chains, and the observed techniques facilitates reporting on observed attacks, making it easier to explain what happened, where it was stopped, and what could have happened next to senior leaders within an organisation. It, similarly, aids in the story-telling process, allowing for repeatable and defined explanations. The BLADE Framework also facilitates an enhanced understanding of the return on investment (ROI) of defensive measures and determining where future investment can best be used.

This more granular and kill chain-based approach to interpreting and understanding these business logic attacks facilitates the adoption of a "defence-in-depth" security model. The historical tendency for many organisations has been to implement a single layer bot management solution (to detect and prevent automated activity) or depend on a Web Application Firewall (WAF) solution to detect large-scale suspicious traffic. By instead adopting a layered-defensive model we are better able to detect and mitigate against business logic attacks. Should an attack bypass the single layer of the single layer model then the attack will be successful. It is thus relatively easy for adversaries to remain ahead of defenders.

Conversely, by adopting a multi-layered defence an adversary who bypasses one layer of defence still has more defensive layers to overcome before the attack can be successful. Mapping detection and mitigation to specific tactics and techniques also allows for a more granular approach.

Rather than looking for technical information that may indicate malicious activity or the employment of a specific tool (such as known bad IPs, user agents, etc.) we can instead look for indicators of a particular sub-technique, technique, or even tactic. This forces an adversary to change their methodological approach, a much larger ask than just altering tooling or infrastructure. This all raises the barrier for entry to adversaries and reduces the ROI from their attacks. In this manner adversaries can initially be dissuaded from launching an attack due to an inability to return sufficient profit from it, and, should they not be dissuaded, be more easily detected and stopped.

THE BLADE FRAMEWORK

The BLADE Framework is intended to be an open-source standard for interpreting and understanding business logic attacks. The name 'BLADE Framework' is taken from "**B**usiness **L**ogic **A**ttack **D**efinition **Framework**".[7] It provides a framework into which all business logic attacks can fit. Not all attacks will go through every stage, and neither will all attacks use all techniques. Rather, all attacks will employ at least one of the tactics, techniques, and/or sub-techniques listed below. This allows any business logic attack kill chain to be mapped to the framework. The framework in its latest iteration can be found at: www.bladeframework.org.[8]

The general progression of any given attack is from left to right across the framework. This should not be taken to mean that every attack will go through all 6 tactics in order. It is, instead, a general guide. In practice, certain types of business logic attack will not necessitate going through every stage, nor will all attacks progress consistently through each stage. For example, it is eminently plausible that an adversary will be challenged by CAPTCHA during the Attack Execution stage and so may need to return to the earlier Defence Bypass Tactic before then returning to Attack Execution to complete their activity. The ordering of the tactics then is intended to be a general guideline through the usual steps taken by an attacker but should not be understood prescriptively.

There are six "tactics". A tactic describes what an attacker is attempting to achieve at this stage of the attack. Underneath tactics sit techniques, each of which describes the methodology an attacker employs in trying to obtain their tactical objective. The sub-techniques are a more granular description of different ways to perform the overarching technique under which they sit. In effect, the tactics are 'why' an adversary is undertaking a particular action, while the techniques and sub-techniques detail 'how' they perform that action.

It should also be noted that many business logic attacks employ bots (i.e., automated tools) to undertake one or more of the techniques in the course of an attack. This may involve full automation of an entire kill chain, or automation of just one action, or any degree of variance in between. When analysing an attack then, anyone employing this framework should be aware that certain actions and techniques may be performed either manually or automatically and be aware of how the use of automated or manual methods may manifest differently in datasets.

This framework is designed to be a 'living' resource. It will be undergoing constant review (at least biannually) and as new tactics, techniques, and sub-techniques are observed in the wild.

The full framework can be found at www.bladeframework.org, the authors would encourage anyone with an interest in bot attacks to review the site.

Footnotes:

7. Although Netacea sponsored the creation of this framework, it is designed to be open-source and available to any who may wish to refer to it. It also does not adhere to Netacea's product offerings, neither covering all of them, nor being entirely covered by them. Instead, it is intended as an independent project separate from the organisation.

8. The BLADE Framework is updated at least biannually by a cross-industry team of core contributors (with urgent updates being implemented ad hoc as required by the changing landscape). Those wishing to provide feedback, propose alterations, or join the core contributions team can do so by following the contribution guidelines on the website.