Riverside: Dynamic Visualization of Network Traffic for Situation Awareness in Computer Security

Kaitlyn DeValk[1]* Niklas Elmqvist[2,1]†

[1]Department of Computer Science and [2]College of Information Studies
University of Maryland, College Park

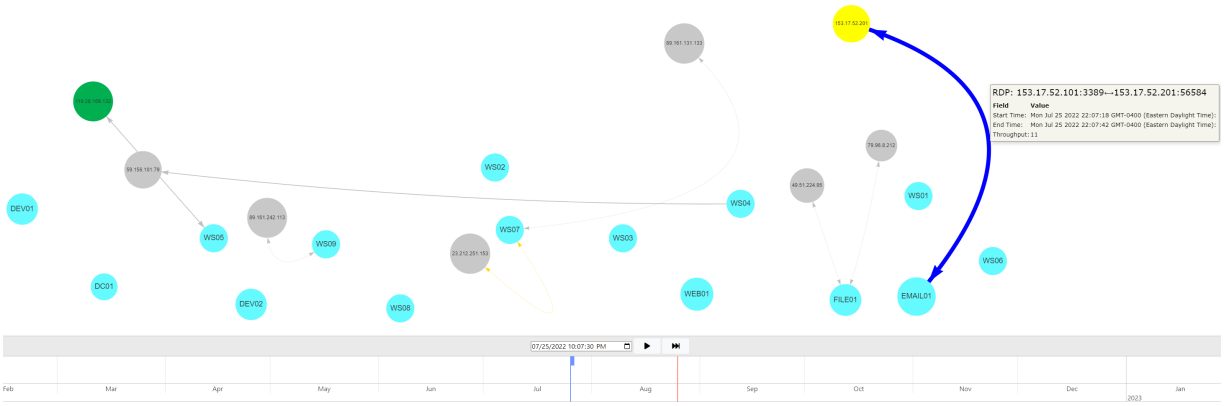

Figure 1: **Overview of the Riverside network visualization tool.** Snapshot in time of the network state showing communication by internal (agents; cyan nodes) and "remote" hosts (green, yellow, gray). Node colors can be changed, like the green and yellow-colored nodes shown above. Edges are automatically colored based on their network protocol. Visualization components can be hovered over to show more information about a node or segment of network traffic, like the RDP traffic displayed. The timeline at the bottom shows both real time (red cursor) and user-specified time (blue cursor), allowing a user to dynamically navigate their network through time.

Abstract

Monitoring security in a computer network requires understanding both real-time network traffic as well as the evolving structure of the network itself. While visualization is increasingly being used for this purpose, current network security tools rely mostly on static network topology and fail to account for user needs. To better understand this problem domain, we interviewed 24 network and security analysts to gain insight on their practices, needs, and current tooling. Based on their qualitative feedback, we designed and built Riverside, a computer security tool visualizing dynamic network traffic across time. By enabling an analyst to navigate traffic in time, summarize intervals, and highlight specific events to prevent change blindness, Riverside gives network and security professionals increased situation awareness of their network.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Graph drawings; Security and privacy—Network security;

1 Introduction

Monitoring network activity can be overwhelming for an analyst, and oftentimes they lose sight of the bigger picture. Visualizations have been shown to help, but many modern security visualizations are based on a static network topology or contain complicated visuals that overwhelm and intimidate users. Furthermore, few tools or papers incorporate users in the design process [5], which leads to less effective visualizations and may explain why many of these tools are not adopted in practice.

In this paper, we conducted interviews with 24 network and security industry professionals to gather their experiences in network security situation awareness. Based on these interviews, we propose Riverside, a dynamic network security visualization tool that provides users with a real-time view of their network as well as high-level insights into their network topology over time. Riverside uses animated node-link diagrams to represent hosts connected by arrows to show traffic flow with the ability to move in time and observe network topology changes.

2 Related Work

Best et al. [4] conducted user studies to understand the needs of industry experts and the challenges with designing useful visualizations for them. Previous interview studies were focused on the evaluation of a tool or the challenges of developing all-encompassing cybersecurity visualization tools. While dynamic network visualizations are not new, many of these techniques have not been applied to cybersecurity visualizations. We take direct motivation from previous work that looks at various ways to develop dynamic graph visualizations [3] but specifically ones that use node-link diagrams to show dynamic timeseries data. Riverside builds on this work by providing animated node-link diagrams and tackles the online and transition problems to offer both internal and external insights over time. Instead of explicitly using staged animation like GraphDiaries [2], Riverside uses node transparency and edge thickness to assist with transitions and change blindness.

Other security situation awareness tools have been developed but lacked tangible network topology layouts and often relied on

*email: ksdevalk@umd.edu
†email: elm@umd.edu

complicated visuals to convey information. Ocelot [1] provides a user-centered design that uses node-link diagrams with circle packing that highlights nodes using a timeline mapping, whereas Riverside provides network snapshots over time and uses animation to display the network. Tools like Gephi can be used to create network visualizations but have additional requirements that can hinder operational deployment.

Other open-source tools also focus on building out a network map but don't allow user input to change the style of the visualization. Unlike previous dynamic visualizations, Riverside allows a user to change the appearance and location of nodes to aid in their spatio-temporal analysis. Additionally, Riverside's focuses on displaying the active network communication vice other tools like Ocelot that select all links and nodes active during a given time window, making it hard to discern necessary details about a network. Riverside gives analysts an explicit network topology at an instant in time which is crucial for a security analyst when trying to pinpoint potentially malicious network traffic or building out an attack timeline.

3 Interview Study

Upon receiving approval from the University of Maryland IRB, we conducted unpaid semi-structured interviews on Zoom video conferences with 24 professionals who had at least one year of experience as a network or security analyst. Interviews lasted 30 minutes and contained questions about their current tooling, professional practices, and experiences with network visualizations.

We performed qualitative coding and thematic analysis, resulting in 14 themes representing experiences participants had about their capabilities, industry truths, challenges participants face, and applicability of data visualization in the security industry. Some of the specific themes that helped drive Riverside's use cases were "Visualizations are not the end all be all," "People like the option to manage their tools," and "Data doesn't lie." With this in mind, we developed a data-driven, network security tool that allows visual changes based on a user's preferences. Furthermore, many of our participants stated that visualizations aren't the only capability they can rely on, and that visualizations should rather provide information that can be used to help them pivot or supply a picture that can be used to explain what an analyst's actions to non-technical personnel.

We also coded 24 features that participants stated would be useful in a novel network visualization tool better suited for their needs. While I haven't been able to incorporate every feature participants mentioned, I did use many of them to guide Riverside's design. Some of the most basic components were the ability to have an infinite canvas and choose what was in the the user's view with the ability to move components of the visualization as they saw fit, provide a big picture view to start with the ability to get more detailed information, and visualize network communication automatically while showing the amount of network traffic. We've been able to incorporate some of these features in full, whereas others are only partially implemented, as this is the first rendition of Riverside.

4 The Riverside Visualization Tool

Riverside is a web-based, dynamic network security visualization tool where animated node-link diagrams are used to show traffic flow over time (Figure 1). We chose node-link diagrams because they provide a concrete view of a network, which was reinforced by our interview participants. While using node-links is the norm, our tool differs from others by offering both real-time and past network states supplemented by user-customization. Some example use cases we identified for Riverside are an incident response team who wants to correlate network changes with alerts over time for an incident, or a security analyst who wants to obtain a network situation awareness tool to supplement their current capabilities.

Riverside uses an agent-server model, where agents are installed on internal hosts and forward network flow to a central server which

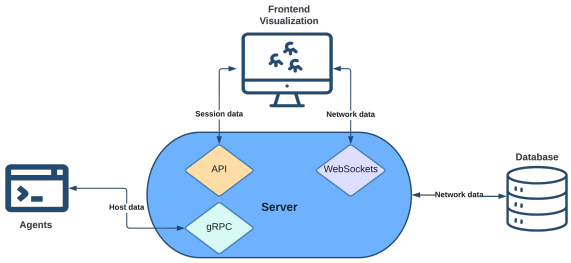

Figure 2: Riverside system architecture.

is stored in a database. The front-end visualization updates through WebSockets with batched real-time messages and is built primarily with the vis.js library. The visualization can be paused, and the timeline can be dragged to see previous communication. It can then play from that point in time, or fast-forward to the current time. Riverside's system architecture is shown in Figure 2.

Agent-installed hosts are always displayed on the visualization with "remote" nodes displayed as they communicate over time. Network flow is shown by the edges that connect nodes on the visualization. The amount of communication is tracked by throughput and displayed by scaling the thickness of the edge. Additionally, if a user hovers over a visualization component, more detailed information is provided about that traffic or host. Node labels, colors, and shapes can also be changed by a user if they wish to differentiate a specific host. Riverside addresses change blindness by making animations, or network communication, more transparent over time if they are a frequent connection. This allows analysts to not be overwhelmed by the differences in the graph over time and focus their attention on new network traffic as they investigate.

5 Conclusion and Future Work

We have introduced a network security situation awareness tool that provides analysts dynamic insights into their network topology over time. We plan to incorporate further functionality, such as more timeline navigation options, filtering capabilities, and node-staining to showcase potential attack paths. Circle packing and clustering could also be used in the future to help with scaling Riverside for larger networks and to visually display network segmentation. Last, we plan to conduct a usability study working with real-world users of Riverside to evaluate and gather feedback for future improvements.

References

[1] D. L. Arendt, R. Burtner, D. M. Best, N. D. Bos, J. R. Gersh, C. D. Piatko, and C. L. Paul. Ocelot: User-Centered Design of a Decision Support Visualization for Network Quarantine. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security*, p. 1–8, 2015. doi: 10.1109/VIZSEC.2015.7312763

[2] B. Bach, E. Pietriga, and J.-D. Fekete. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, May 2014. doi: 10.1109/TVCG.2013.254

[3] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The State of the Art in Visualizing Dynamic Graphs. *EuroVis (STARs)*, 2014. doi: 10.2312/eurovisstar.20141174

[4] D. M. Best, A. Endert, and D. Kidwell. 7 Key Challenges for Visualization in Cyber Network Defense. In *Proceedings of the IEEE VIS Workshop on Visualization for Cyber Security*, p. 33–40, 2014. doi: 10.1145/2671491.2671497

[5] S. McKenna, D. Staheli, and M. Meyer. Unlocking User-Centered Design methods for Building Cyber Security Visualizations. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security*, p. 1–8. IEEE, 2015. doi: 10.1109/VIZSEC.2015.7312771